

AD-A052 900

SCIENCE APPLICATIONS INC ARLINGTON VA

F/G 14/5

IMAGE UNDERSTANDING PROCEEDINGS OF A WORKSHOP HELD AT MINNEAPOL--ETC(U)

APR 77 L S BAUMANN

F30602-76-C-0165

UNCLASSIFIED

SAI-78-549-WA

NL

1 OF 2  
AD  
A062900







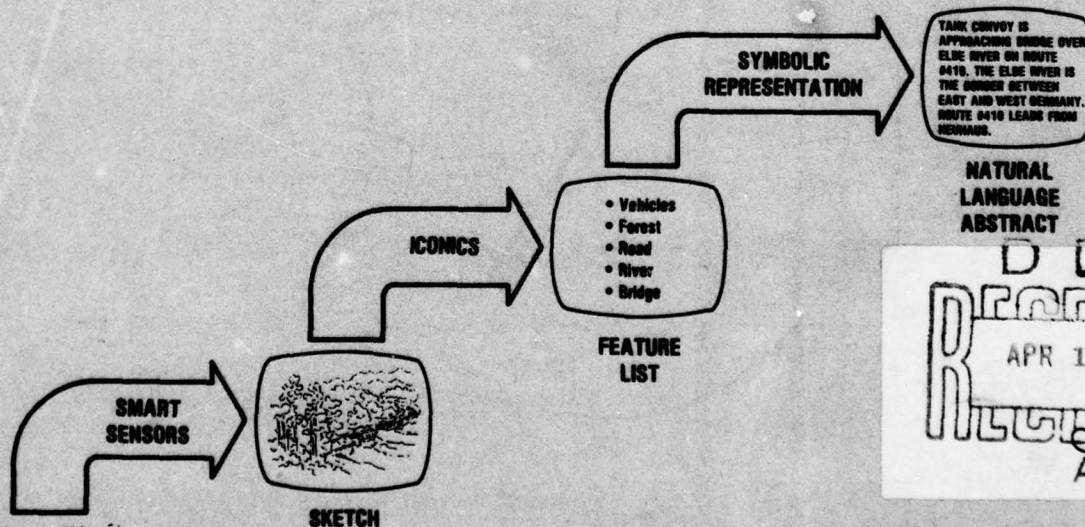
AD A 052900

# PROCEEDINGS: IMAGE UNDERSTANDING WORKSHOP

April 1977

Sponsored by:  
Information Processing Techniques Office  
Defense Advanced Research Projects Agency

AD No. 1  
DDC FILE COPY



NATURAL  
LANGUAGE  
ABSTRACT

DDC  
RECEIVED  
APR 19 1978  
AV

## DISTRIBUTION STATEMENT A

Approved for public release;  
Distribution Unlimited

This report was prepared by  
Science Applications, Inc. for  
the Defense Advanced Research  
Projects Agency under Contract  
F 30602-76-C-0165

022

1

6

# IMAGE UNDERSTANDING

Proceedings of a Workshop  
held at  
Minneapolis, Minnesota,  
April 20, 1977,

Sponsored by the  
Defense Advanced Research Projects Agency

11

11 Apr 77

12 12 pp.

14

Science Applications, Inc.  
Report Number SAI-78-549-WA  
10 Lee S/Baumann  
Workshop Organizer and  
Proceedings Editor

DDC  
RECEIVED  
APR 19 1978  
A

15

This report was supported by  
the Defense Advanced Research  
Projects Agency under Contract  
F30602-76-C-0165 monitored  
by the Rome Air Development  
Center, Griffiss AFB, N.Y.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of the Defense Advanced Research Projects Agency or the United States Government.

1

APPROVED FOR PUBLIC RELEASE  
DISTRIBUTION UNLIMITED

407 154

Gu



## TABLE OF CONTENTS

### SESSION I

UNDERSTANDING SYSTEMS AND KNOWLEDGE REPRESENTATION  
Moderator: Dr. William A. Sanders - Army Research Office

SPATIAL UNDERSTANDING	
R. Arnold, Stanford University.....	1
REPRESENTATION AND USE OF KNOWLEDGE IN GOAL-DIRECTED VISION SYSTEM	
C. Brown and K. Lantz, The University of Rochester.....	5
THE LOCUS MODEL OF SEARCH AND ITS USE IN IMAGE INTERPRETATION	
S. Rubin and R. Reddy, Carnegie-Mellon University.....	12
COOPERATIVE COMPUTATION OF STEREO DISPARITY	
D. Marr and T. Poggio, Massachusetts Institute of Technology.....	15

### SESSION II

IMAGE REGISTRATION AND RECOGNITION  
Moderator: LTC George W. McKemie - Air Force Office of  
Scientific Research

PARAMETRIC CORRESPONDENCE AND CHAMFER MATCHING: TWO NEW TECHNIQUES FOR IMAGE MATCHING	
H. Barrow, J. Tennenbaum, R. Bolles and H. Wolf, Stanford Research Institute....	21
SYMBOLIC IMAGE REGISTRATION AND CHANGE DETECTION	
K. Price and R. Reddy, Carnegie-Mellon University.....	28
IMAGE SEGMENTATION AND OBJECT DETECTION BY A SYNTACTIC METHOD	
J. Keng, Purdue University.....	38

### SESSION III

Moderator: Mr. John S. Denhe - Army  
Night Vision Laboratory

A BOTTOM UP IMAGE SEGMENTOR	
G. Coleman and H. Andrews, University of Southern California.....	44
A COMPARISON OF SOME SEGMENTATION TECHNIQUES	
R. Nevatia and K. Price, University of Southern California.....	55
REGION EXTRACTION USING CONVERGENT EVIDENCE	
D. Milgram, University of Maryland.....	58
SEGMENTATION OF FLIR IMAGES BY PIXEL CLASSIFICATION	
D. Panda, University of Maryland.....	65

### SESSION IV

SIGNAL PROCESSING, SOFTWARE AND HARDWARE  
Moderator: Dr. John J. Knab - Air Force  
Avionics Laboratory

IMAGE SEGMENTATION USING TEXTURE AND GRAY LEVEL	
S. Carlton and O. Mitchell, Purdue University.....	71
SYMBOLIC ANALYSIS OF IMAGES USING PROTOTYPE SIMILARITY	
R. Touchberry and R. Larson, Honeywell, Inc.....	78
SYSTEM SUPPORT FOR A DISTRIBUTED IMAGE UNDERSTANDING PROGRAM	
J. Feldman and R. Rashid, University of Maryland.....	83

ACCESSION IN	
RTIG	Write Section <input checked="" type="checkbox"/>
DDG	Diff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and/or SPECIAL
A	

*in*

# TABLE OF CONTENTS (Cont.)

AUTOMATIC TARGET CUEING ON THE FOCAL PLANE	
T. Willett and N. Bluzer, Westinghouse Systems Development Division.....	87
CCD IMAGE PROCESSING CIRCUITRY	
G. Nudd, Hughes Research Laboratories.....	89

## SESSION V

### PROGRAM REVIEWS BY PRINCIPAL INVESTIGATORS

Chairman: Major D. Carlstrom, DARPA

IMAGE UNDERSTANDING RESEARCH AT CMU: A PROGRESS REPORT	
R. Reddy, Carnegie-Mellon University.....	95
1976-77 PROGRAM REVIEW	
R. Larson, Honeywell, Inc.....	96
ALGORITHMS AND HARDWARE TECHNOLOGY FOR IMAGE RECOGNITION -- PROJECT STATUS	
REPORT - MARCH, 1977	
A. Rosenfeld, University of Maryland.....	98
IMAGE UNDERSTANDING AND INFORMATION EXTRACTION	
K. Fu and T. Huang, Purdue University.....	101
OVERVIEW OF THE ROCHESTER IMAGE UNDERSTANDING PROJECT	
J. Feldman, University of Rochester.....	103
INTERACTIVE AIDS FOR CARTOGRAPHY AND PHOTO INTERPRETATION: PROGRESS REPORT	
APRIL 1976 TO APRIL 1977	
H. Barrow, Stanford Research Institute.....	105
THE USC IMAGE UNDERSTANDING PROJECT, 1 OCTOBER 1976 TO 31 MARCH 1977	
H. Andrews, University of Southern California.....	109

Material for the Program Review from P. Winston of M.I.T. and T. Binford of Stanford University was not available for inclusion in this document.

*iii*

## FOREWORD

The Image Understanding Program is planned to be a five year research effort to develop the technology required for automatic and semiautomatic interpretation and analysis of military photographs and related images. This program, now in its second year of Defense Advanced Research Projects Agency (DARPA) sponsorship, was initially funded in 1976.

This document contains papers submitted by various research personnel working on projects in the Image Understanding Program. These papers were presented on April 20, 1977 at the fifth Image Understanding Workshop held in Minneapolis, Minnesota. The Workshop was hosted by Dr. T. F. Hueter, Vice President for Corporate Technology, Honeywell, Inc.

The current DARPA program includes four University/Industrial teams:

- University of Southern California - Hughes Research Laboratories
- University of Maryland - Westinghouse, Inc.
- Purdue University - Honeywell, Inc.
- Carnegie-Mellon University - Control Data Corporation

There are also five individual DARPA-sponsored research efforts included:

- Massachusetts Institute of Technology
- Stanford University
- University of Rochester
- Stanford Research Institute
- Honeywell, Inc.

The purpose of the workshop was to enable various program researchers to present interesting technical accomplishments achieved during the past six months. The status of each of the diverse projects including future research plans and goals were also agenda objectives. In this way, by stimulating cross-fertilization discussions, it was hoped to assist community-wide understanding of the individual research efforts. Since the participants included personnel from the military research and development community, as well as representatives from interested user organizations, the workshop served as a means to provide a "dialogue" between researcher and user. Such information exchange is considered a must by DARPA management in order to facilitate technology transfers.

The workshop was organized into four sessions which ranged from the broadest applications down to more specific investigations. Each principal investigator presented his program for review. A general discussion period open to all participants was conducted following the presentations.

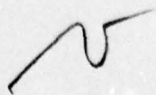
The Image Understanding Program is under the direction of Major David L. Carlstrom, USAF, of the Defense Advanced Research Projects Agency (DARPA), Information Processing Techniques Office. The cover design of this document was taken from a diagram used by Major Carlstrom to explain the hierarchical processing required to convert basic image data into real-time information for decisionmakers. Major Carlstrom has repeatedly reminded researchers that this end result must be clearly kept in mind as finite improvements are achieved at each level along the way.

*iv*



The conference organizer wishes to thank Dr. William A. Sander of the Army Research Office; LTC George W. McKemie of the Air Force Office of Scientific Research; Mr. John S. Denhe of the U. S. Army Night Vision Laboratory; and Dr. John J. Knab of the Air Force Avionics Laboratory for acting as moderators for the technical sessions. Also, Mr. Rod Larson and Ms. Beverley Jensen of Honeywell, Inc. were most instrumental in the conduct of the workshop by securing facilities, making arrangements and generally assisting in the coordination necessary to provide for the needs of the participants. Typing support and collection and arrangements of papers was accomplished by Ms. Gloria Wilkie of Science Applications, Inc.

Lee S. Baumann  
Science Applications, Inc.  
Workshop Organizer



# AUTHOR INDEX

<u>Name</u>	<u>Page No.</u>	<u>Name</u>	<u>Page No.</u>
ANDREWS, H.C.	44-54, 109-113	MARR, D.	15-20
ARNOLD, R.D.	1-4	MILGRAM, D.L.	58-64
BARROW, H.G.	21-27, 105-108	MITCHELL, O.R.	71-77
BINFORD, T.O.	1-4	NEVATIA, R.	55-57
BLUZER, N.	87-88	NUDD, G.R.	89-94
BOLLES, R.C.	21-27	PANDA, D.P.	65-70
BROWN, C.	5-11	POGGIO, T.	15-20
CARLTON, S.G.	71-77	PRICE, K.	28-31, 55-57
COLEMAN, G.	44-54	RASHID, R.	83-86
FELDMAN, J.	83-86, 103-104	REDDY, R.	12-14, 28-31, 95
FU, K.S.	101-102	ROSENFELD, A.	98-100
GENNERY, D.B.	1-4	RUBIN, S.	12-14
GLISH, B.	32-37	SWANLUND, G.	32-37
HUANG, T.S.	101-102	TENENBAUM, J.M.	21-27
KENG, J.	38-43	TOUCHBERRY, R.	78-82
KOBER, W.	32-37	WILLETT, T.	87-88
LANTZ, K.	5-11	WOLF, H.C.	21-27
LARSON, R.	78-82, 96-97		

*vi*

## SPATIAL UNDERSTANDING

R.D. Arnold  
T.O. Binford  
D.B. Gennery

Artificial Intelligence Laboratory, Computer Science Department  
Stanford University, Stanford, California 94305

## Abstract

We present recent progress in stereo photointerpretation applied to vehicle location and building location: segmentation of vehicles from ground, and preliminary description of their shape preliminary to identification of vehicles in aerial images of suburban scenes; segmentation of vehicles from ground in ground level images; preliminary results in segmentation and description of buildings in aerial images; a new technique for feature-based stereo in which edge fragments are linked into smooth curves in 3d; depth mapping based on area correlation.

## Introduction

Our goal has been to develop techniques for passive ranging in PI and guidance using sequences of images from a moving observer. We have two other goals: first, to describe and identify objects seen from a variety of viewpoints, in this case aerial and ground views; second, to use edge features in order to exploit the ARPA "smart sensor" technology, and to increase the accuracy with which measurements can be made.

Our program is called Spatial Understanding. The aim of the program is to build descriptions which are segmented into surfaces and volumes, and to match segmented spatial descriptions. In terms of interactive aids to photointerpretation, the importance of this approach is that it makes use of stereo, and that its representations are intuitively natural to humans. Natural representations are essential to our next phase of interactive programming of PI tasks.

At the last image Understanding workshop, we presented results on finding the ground surface in aerial images. We summarize those results as an introduction to the current research. The system starts with a sequence of images. The system first orients itself by finding an Observer Model for the sequence of images. It uses the Observer Model to: limit search in subsequent ranging; calculate range to image points; to guide itself toward a target, or away from obstacles.

Building the Observer Model takes at least half the total time in finding the ground surface. The Observer Model can be calculated from accurate guidance information, thus eliminating half the computation. Any guidance information helps. In

most cases, enough guidance information and knowledge about the scene is available to cut the computation time by large factors from those quoted below.

The system first finds a small sample of interesting features in one image and matches them with their corresponding view in the other image. The Observer Model can be found from 5 features which are non-degenerate. Typically, 10 features are used because some feature matches may be wrong and some sets of features may be degenerate. Interesting features are small areas which can be localized in two dimensions without an Observer Model. Those are features which are not invariant along any direction. Lines are not localizable, but corners are.

A 2d search is necessary to match features without an Observer Model. An important contribution is the binary search correlation algorithm which finds matches anywhere in the image in only 50 msec. It uses a coarse-to-fine binary search strategy: it first searches on a coarse version of the image (16x16); then it searches in higher resolution images (32x32, 64x64, etc), each time in the neighborhood of the best match in the previous image. It has an error rate of about 10% false matches in an extensive set of images. It encounters fewer ambiguities than brute force matching since not only must the feature match, but the surrounding context must match also.

The system selects a larger sample of interesting points to find corresponding views in the two images. It then finds a ground surface (quadratic) for small portions of the scene. The ground surface finder maximizes a function which favors as many points as possible near the surface and as few points as possible below the ground surface.

Approximate timings for these modules on a PDP KL/10 are:

Interest operator	75 msec
Binary search correlator	50 * N1 msec
Observer Model solver	250 * N1 msec

where N1 is the number of points in the sample used to determine the Observer Model (typically 10).

$T_1 = 75 \text{ msec} + 300 * N1 \text{ msec}$  (typically 3.1 sec) where  $T_1$  is the time required to determine the Observer Model. Solution of the Observer Model



requires 80% of the computation time required for obtaining the Observer Model. It is expected that the solution can be speeded up a factor of 5 (although it has been carefully constructed with concern for efficiency). That would make T1 typically 1.1 sec.

Timings for the ground surface finding are:

Range Sample 50 \* N2 msec

Ground Surface solver 5 \* N2 msec

where N2 is the number of range points in the sample, typically 50.

$T2 = 55 * N2$  msec (typically 2.7 sec)

where T2 is the time required to determine the ground surface model, given the Observer Model.

Finding the ground surface model can be speeded up. The Observer Model makes it economical to make a range map. A point in one view corresponds to a ray in space, which corresponds to a line in the other view. Nearby points usually have approximately the same disparity. Thus, search can be limited to a small interval on a line. We have estimated that about 2 msec per point are necessary for this search, which would make  $T2 = 7 * N2$  msec (typically .35 sec). For many missions in PI or guidance, enough information will be available from instrumentation or inertial guidance to eliminate determination of the Observer Model, and to allow frame to frame tracking times less than a second on an ordinary computer. We estimate that the computation time will be approximately twice as long on a PDP11/45.

Ground surface finding should work for images from a variety of sensors and including contrast reversal. The binary search algorithm used in obtaining the range sample for the Observer Model should be successful where depth differences in the scene are small compared to the range (almost always true in aerial images) and images are similar (not true for different sensors). Binary search probably will fail at the coarse stage in dissimilar parts of scenes. It is possible to instead match curves in images, using techniques developed here. Curve matching should be effective even in these cases. Alternatively, guidance information may be used to obtain the Observer Model in those cases.

#### Range Mapping

A routine now maps ranges over whole images, using area correlation. It has made a dense map of ranges in a pair of images of vehicles in a parking lot, taken from near ground level. The range map was used as input for the ground plane finder. Figure 1 shows one of a pair of images. Figure 2 shows the heights of areas which are at least two feet above ground level. These areas coincide with the two cars in the images, and heights are reasonably accurate.

The routine uses a high-resolution correlator to obtain as much accuracy as possible. The correlator calculates a probability of match, interpolates to the best match, and calculates position errors in match. The ground plane finder uses these estimates of position errors. The range

mapper accepts matches only if they satisfy reasonable probability of match, and if a neighbor matches at similar disparity. The mapping routine uses both the Observer Model and continuity of depth surfaces to limit search. It searches initially near neighboring matches; when necessary, it carries out a full search along the ray from minimum distance to infinity.

#### Car Location

We have developed a new technique for edge matching and curve linking in stereo. Ranging based on matching edge features increases the accuracy of determining boundaries of objects by a factor of about 20. This makes it possible to use fairly accurate estimates of object size. Edges also provide additional information about surface markings which are not available in stereo based on area correlation.

The technique has been used for segmentation and description of vehicles in aerial images. Figure 3 shows part of an image from a pair of a parking lot. Figure 4 shows edges 2 feet or more above ground level, in a coordinate system with x axis along the stereo baseline. Edges have been linked together and fit with straight lines. Rectangles have been fit to the vehicles, with approximately the right orientation and size. We expect the system to identify cars soon.

Edge elements (edgels) are linked into smooth curves in 3d. Not only must they link up in the image, but they must be continuous in disparity also. The matching and linking process makes use of the Observer Model and ground surface model which are already determined. It first transforms edgels to a standard stereo system ( $x'$ ,  $y'$ ) with the baseline along the  $x'$  axis. Edges in Figure 4 are shown in the stereo coordinate system. The display is distorted slightly because of the aspect ratio of the display. Each edgel is put into a cell in an ( $x'$ ,  $y'$ ) grid. Each cell is 8x8 pixels square in ( $x'$ ,  $y'$ ); it contains a list of edgels in the cell. View 1 is scanned cell by cell. For each edgel, the routine looks at all edgels in view 2 with permissible disparities. It ignores edgels near the stereo axis (within 25 degrees). It rejects any pairs which are more than 1 radian apart in angle. The routine could make use of special knowledge about horizontal edges to make tight limits on angle. That special case is useful for vehicles and for buildings. The routine also requires that pairs match in contrast (sum of squares of signal) and brightness. It picks the best match.

Then the routine makes another pass through the array. It looks in a 16x16 pixel area around each edgel in view 1. It checks to see whether neighbor edgels are colinear and compatible in contrast and brightness, and whether they match in view 2 with consistent disparity. If there are 2 neighbor edgels which link in this way, then the edgel is accepted. A line is fit to the list of linked edges.

Linked edges are given to the rectangle finder. It

finds the maximum of a histogram of edges versus angle module 90 degrees to find the orientation, theta, of the rectangle. It computes transformed coordinates in a system (u,v) rotated along theta. Then it finds clusters in histograms with respect to u and v separately. It takes all combinations of clusters to compute the product of all probabilities for that rectangle. The combination with the max probability is assumed to be the answer. It calculates the greatest lower bound in cases in which the rectangle is not bounded along one side, and uses default information on length and width where necessary.

## Car Models

Cars are modeled by planes and boxes. Planes and edges are nearest to observable. There are vertical planes and horizontal planes. The box model of cars consists of two boxes, one on top of the other. The sizes of the boxes have a relatively small range. The upper box is at an approximately constant location from the front of the lower box. In some cases, this enables distinguishing front from rear. The approximate dimensions of the upper box are: height 56", width 60" and length 80". It has a horizontal top and vertical sides. The lower box has dimensions: height 36", width 60" and length 160".

## Buildings

The same techniques are being used to segment buildings from ground, to model the segmented objects, and to form building models. We expect the techniques to work better. Buildings are larger, they are more planar. We expect to present preliminary results in description of buildings.

Figure 1  
Ground Level View of Parking Lot

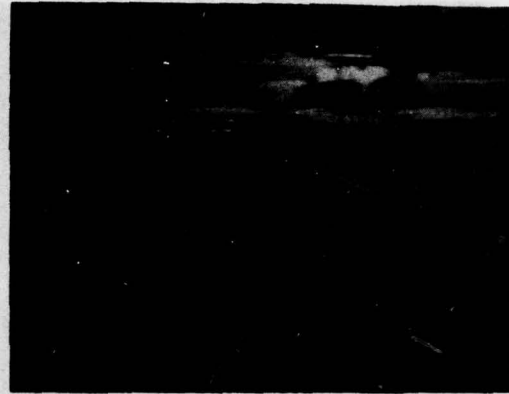


Figure 2  
Height of Areas 2ft. or More Above Ground Sur  
Surface (From Figure 1)

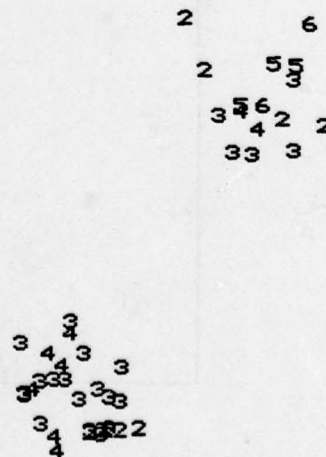


Figure 3  
Aerial View of Parking Lot

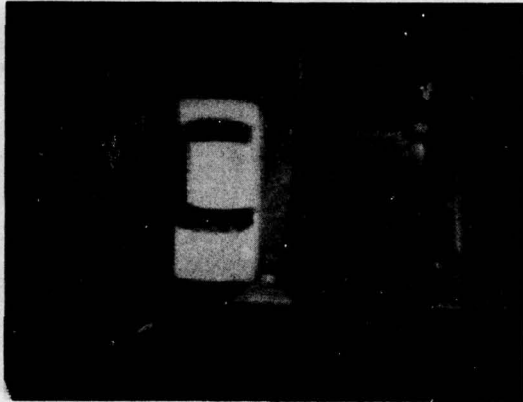


Figure 4  
In Stereo Coordinate System  
Left: Linked Edges 0-2ft. Above Ground  
Right: Linked Edges 3-6ft. Above Ground

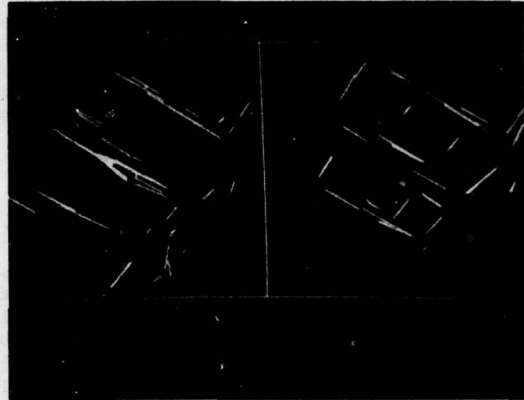
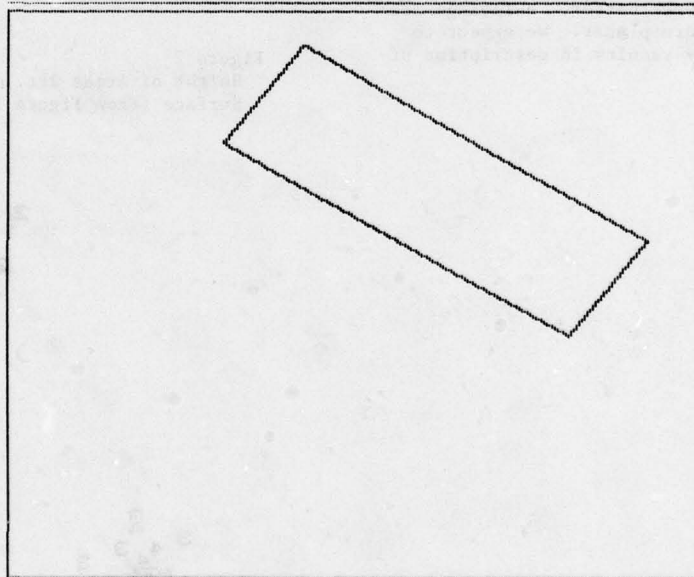


Figure 5  
Rectangle Fit to Upper Car  
In Stereo Coordinate System





# REPRESENTATION AND USE OF KNOWLEDGE IN A GOAL-DIRECTED VISION SYSTEM

C. Brown and K. Lantz

Computer Science Department  
The University of Rochester

## ABSTRACT

A vision system is described which is geared to:

- extracting information efficiently at variable levels of detail;
- allowing users to bring to bear specialized knowledge about strategies, representations, and techniques;
- representing and using general and map-derived knowledge in semantic net form;
- providing nonstandard control structure, rudimentary automatic inferencing, and facilities for automatic procedure selection.

An illustration is provided by current work in ship-finding.

- returning to an aerial photo on different days to perform different tasks;
- generating different special-purpose maps from the same photo.

For this approach to image analysis, it is important to define a representation that allows extensions to partial mappings which may be known a priori or acquired sequentially. For efficiency, we want a way of defining quantitatively when the query has been satisfied so that we do not perform unnecessary mappings.

Queries will initially be made by writing programs, but the system will not consist of one or more monolithic programs that rigidly solve single specialized problems. Rather, a central concern is to develop the idea of standard representations for common low- and high-level objects so as to facilitate communication between procedures. Standard representations, if they can be found, will explicate some primitive constructs useful in vision, and will make vision programming easier. If in addition the representations are machine-interpretable, then the programs can begin to monitor, reason about, and affect their own performance. An important object in the system is the procedure. Procedures are often attached to objects (a "how to find it" procedure, for instance), and an automatically-interpretable description of the actions and characteristics of procedures may be used to choose automatically the most reliable, cheap, or accurate procedure for a given job. Also, such descriptions allow for incremental, modular extensions to the power of the system without any reprogramming (see Section 3.4.1).

## 1. INTRODUCTION

For some time we have been developing a general vision system. Sections 1, 2, and 3.4.1 are condensed from [Ballard, Brown and Feldman], in which more detail may be found. The system is structured in three layers. At one end of the structure is a semantic network, the world model; it contains idealized prototypes of structures from low level (such as edges) to high level (e.g. complex assemblages of world objects). In the middle we have a sketchmap. This data structure is synthesized during image analysis and provides a mapping between the model and the image. At the other end is the image data structure, consisting of the original image and various processed versions of it.

The concept of a query is central to our approach to image analysis. Given a richly descriptive image model, a query in the form of a special-purpose program can be coded in such a way as to require mapping a minimum of model structure into the image. Another aspect of the design is the retention and use of information gained in previous tasks so that mappings may be refined over a succession of queries. Examples might be:

## 2. SOME GENERAL ASPECTS OF THE SYSTEM

### 2.1. MODEL STRUCTURES

In a query-oriented system, one does not always want to perform an exhaustive initial segmentation of the scene into regions, line segments, or anything else. Such segmentation may be at a level of detail which is too coarse or too fine to reveal what one wishes to know. Further, even when segmentation is data-directed, a uniform algorithm producing a continuum of

intermediate data structures may be too inflexible and (given present understandings) inefficient.

We desire to replace mandatory processing through many levels of detail by modelling objects at many useful levels of detail, and using procedures capable of selecting different image resolutions. The supposition is that the purposes of the query will stop the analysis at the minimum necessary level of development.

The model holds several kinds of knowledge about the image domain (see Section 3.2). It includes a relational network of nodes which are identifiable with (primitive and complex) objects and concepts in the domain from which the scene is taken. The answer to a query is a sketchmap consisting of instantiations of model nodes. The model, therefore, contains knowledge in the form of all potential instantiable descriptions. An example of this kind of knowledge is the assertion "docked ships are adjacent to docks." It is potentially part of the model-image mapping since both "ship" and "dock" could be instantiated with pointers to regions of the image. The model also contains knowledge not in the form of a mappable assertion but still useful in mapping, for instance: "ships are about 6 times as long as wide, and are about 300 feet long."

Model nodes are identifiable with concepts in the scene domain and each has links to other nodes; they have a rich structure (see Section 3.3). Procedures may be attached to nodes to allow choice of control regimes, but we do not envision that the structure will be self-activating.

In the process of synthesizing a model-image mapping, special-purpose procedures generate and use many kinds of knowledge in the form of image data structures, parameters, operators, and descriptions of their results. A structuring of this diverse knowledge is provided by standard data objects which are used for communication between the various knowledge sources and the users.

One example of an important standard object is the location descriptor, which contains what is known of how to locate an entity. One aspect of location description has been called [Bolles] a tolerance region. There are many advantages to having a standard representation for object locations:

- If such descriptions are data types, their computations can be separated from the procedures that use them. If they can be passed as arguments, they provide a certain "common currency" between procedures, thus simplifying and modularizing the procedures that use them.
- Location descriptors can represent approximate locations, which is useful for queries unconcerned with exact answers.

- Constraints between locations can propagate knowledge throughout the model (see Section 3.4.2). Location descriptors can be computed from other location descriptors via relations, or by union and intersection of the described point sets. A system which applied linear programming techniques to the problem of locating regions through constraints placed on their boundaries was developed in [Taylor].
- Use of location descriptors is geared to an abandonment of the exhaustive segmentation paradigm wherein every region must correspond to some object. Different location descriptors may refer to disjoint point sets or may overlap on the image, and different objects may have similar location descriptors.

## 2.2. IMAGE AND MAP STRUCTURES

In the process of analyzing an image many intermediate, processed image data structures will be generated. We plan to attach a description to each of these structures. This will facilitate the writing of large processing procedures in the following ways:

- entire structures can be passed to procedures as arguments;
- intermediate results can be stored in a standard way;
- image transfer through computer networks will be facilitated [Maleson and Rashid].

The system uses a version of the Array of Image Samples format proposed in [Sproull and Baudelaire], extended to contain information from processed and interpreted images as well as stylized, pictorial structures isomorphically related to the image, such as topographic maps.

A "map" is a useful entity for tops-down image analysis; we take a map to be any data structure which contains information about how we expect an image to look. It may have a large amount of metric information, as would the topographic map mentioned above. It may be a finite set of assertions giving purely relational information. Maps may thus give only enough information for a specific purpose.

For topographic maps, the system will use something like an extended GIST format [Lemmer], consisting of point, linear, and area features inverted on feature type, as well as a potential for raster information. Typically only a small subset of a topographic map is relevant to any given image understanding task. The system may need routines for converting (in some directions) between the various representations of maps. Especially useful is conversion of linear features into raster data.



### 2.3. CONTROL

The procedures attached to nodes of the model could be run in the style of active knowledge [Freuder] or in any other nonstandard control regime. These control structures have been used to achieve knowledge propagation, shifts of attention, parallel processing, etc.

Some control will be overseen by the system (see Section 3.4). However, the system is at present not committed to any particular control regime. We are interested in finding out what control primitives are the most helpful, but as yet do not feel strongly enough about any scheme to restrict the user to some system control philosophy. We will be making queries at several levels of detail; we will (initially, at least) code into the query strategies for answering the question at the right level of detail rather than expecting the model to provide them. The model will be considered more as a data structure than as a description of control.

## 3. SPECIFIC ASPECTS OF THE MODEL

### 3.1. REPRESENTATION

Our system is partitioned into model, sketchmap, and image. The image component consists solely of data--the image at various magnifications, resolutions, etc. The model and sketchmap, on the other hand, comprise knowledge about the world; the model represents generic knowledge, whereas the sketchmap is a specific instantiation of a subset of the model. We will express the content of the model with constructs from Knowledge Representation Language (KRL) [Bobrow and Winograd].

In our system the basic object is a node. A node is a referent for an entity or category in the "world" being represented. Nodes have a name (e.g., "shape object," "linear feature," or "ship"), type, and a variable number of associated properties, or slots. A generic node is a prototype; an instantiation thereof is an individual. A slot is effectively a (property, value) pair--for example, the tonnage of a ship is 100000 tons, or its silhouette is given by a point set. Slots may contain procedures to be called as circumstances warrant; they can serve the role of "servants" and "demons" for propagating or acquiring knowledge.

KRL, as formulated by Bobrow and Winograd, is LISP-ish in syntax. In what follows we will employ a syntax based on the fact that our system is written in LEAP [Feldman and Rovner]. Nodes are constructed from associations (triples). The LEAP association

<attribute> of <object> = <value>

or

<slot> of <node> = <value>

encodes a semantic net representation wherein the <attribute> (<slot>) links the two entities <object> (<node>) and <value>. In this discussion we represent a typical prototype node as follows:

```
[ <node-name>
  nodetype:      <node-type>
  isa:           <node>
  son:           <node>
  <slot-name>1 : <value>
  ...
  <slot-name>n : <value> ]
```

A basic activity of image understanding is finding individuals and deducing certain of their properties. When this happens, an "individual" instance of a prototype node is created with just those properties that are relevant to the task at hand. Such "partial instantiations" are easy in LEAP. An individual may be characterized in a natural way as:

```
a <node> with <slot-name>1 = value
...
<slot-name>n = value
```

Generic properties are found by following the isa links through the hierarchy of individuals and prototypes.

### 3.2. STRUCTURE OF THE MODEL

The vision "universe" is composed of the real world, the image, and abstract entities. The world is composed of objects--ships, docks, oceans, roads, posts, etc. At a higher level these can be grouped into area objects (oceans), shape objects (ships), linear objects (coastlines), and point objects (bows of ships); at a lower level the viewer sees particular instantiations of ships, docks, and coastlines--i.e., he sees individuals. The image, on the other hand, is composed of features--lines, edges, points, regions. As with world objects, features can be grouped into area, shape, linear, and point features, and instantiated to yield particular individuals.

To describe the image and the world, and to correlate the two, it is necessary to deal with abstract entities. One such abstract entity is the location descriptor, which relates locations of nodes to coordinate systems, which are themselves abstract entities. Other abstract entities are points, point sets, line segments, coordinate systems, chain codes, etc. Such entities comprise the third fundamental partition of the vision model; many of them represent knowledge concerning relationships between the world and the image. Our basic model structure, then, appears as in Figure 1.

Model

```

World Object
  Area Object ...
  Shape Object
    Ship
      Ship001 ...
    ...
  Linear Object ...
  Point Object ...
  ...
Abstract Entity
  Coordinate System...
  Location Descriptor
    Area Location Descriptor ...
    Shape Location Descriptor ...
    Linear Location Descriptor ...
    Point Location Descriptor ...
  Point ...
  ...
Image Feature
  Area Feature ...
  Shape Feature ...
  Linear Feature
    Line
      Line001 ...
    ...
  Point Feature ...
  ...

```

Figure 1

### 3.3. SHIPS: AN EXAMPLE OF NODE SEMANTICS

In Figure 1, Ship001 is an instance of a prototype Ship. Ship, in turn, isa Shape Object, which isa Object in the world. Objects are fixed in place (in both the image and the world) via location descriptors and might appear as follows:

```

[ Object
  nodetype:      basic prototype
  objecttype:    OneOf {Point, Linear, Shape,
                      Area}
  son:           selection (the objecttype of
                    Object ThisOne) into
                    [Point:: a PointObject
                     Linear:: a LinearObject
                     Shape:: a ShapeObject
                     Area::  an AreaObject]
  worldlocation: selection (the objecttype of
                    Object ThisOne) into
                    [Point:: a PtLocnDescr
                     Linear:: a LinLocnDescr
                     Shape:: a ShLocnDescr
                     Area::  an ArLocnDescr]
  imagelocation: //...similar to worldlocation ]

```

"(the objecttype from Object ThisOne)" is a description of a slot (objecttype) which resides in a particular (Object) node; here, the (Object) node is ThisOne--i.e., the same node the description is in. Given an instantiation of an Object node with objecttype Point, the selection specifies that the son of the instantiated object must

be a PointObject, and the worldlocation and imagelocation are both PointLocationDescriptors.

Shape objects are distinguished by one or more of the properties (not necessarily a complete list) found in the slots of a prototype ShapeObject node:

```

[ ShapeObject
  nodetype:      basic prototype
  isa:           an Object with
                 objecttype = Shape
                 worldlocation = (the
                                   wrldlocn from
                                   ShapeObject ThisOne)
                 imagelocation = (the
                                   imlocn from ShapeObject
                                   ThisOne)
  son:           OneOf {(a Ship), (a Car),
                       ...}
                 //all possible shape objects
  imagelocation: a ShapeLocationDescriptor
  worldlocation: a ShapeLocationDescriptor
  virtualcoords: a CoordinateSystem
                 //virtual coordinate system for
                 remaining slots
  boundary:      a DirectedLineSet
                 //outline in virtual coordinates
  centroid:      a Point
  orientation:    a Number
                 //orientation of "midline" with
                 respect to an axis
  template:      a PointArray
                 //"silhouette" in virtual coordinates
  convexity:      a Number
  aspectratio:   a Number
  length:        a Number
  width:         a Number ]

```

The description "(the worldlocation from ShapeObject ThisOne)" in an Object node's worldlocation slot means that the instantiated Object will be given the same worldlocation as the instantiated ShapeObject without copying information.

A prototype Ship is a specialization of a ShapeObject. It inherits implicitly all properties of a ShapeObject but can possess other distinguishing properties as well--e.g., tonnage, bow, and stern. A ship prototype could be defined:

```

[ Ship
  nodetype:      specialization prototype
  isa:           a ShapeObject
  shiptype:      OneOf {Carrier, Battleship,
                      Cruiser, Destroyer,
                      Tanker, Freighter}
  tonnage:      a Number
  name:         a String
  bow:          a PointObject
  stern:        a LinearObject ]

```

If a ship is found in the image, the sketch-map gets a new node which is an instance of a Ship node and which has some subset of a Ship's properties associated with it. This entails

creating a new instance of a ShapeObject and thus an instance of an Object node. Finally, the 35,000-ton cruiser "U.S.S. Montlantz" located at 21 36'N, 106 47'E might be represented as follows:

```
[ Ship001
  nodetype: individual
  isa:      a Ship with
            isa = ShapeObject001
            shiptype = Cruiser
            tonnage = 35000
            name = "U.S.S. MontLantz" ]

[ ShapeObject001
  nodetype: individual
  isa:      a ShapeObject with
            isa = Object001
            son = Ship001
            worldlocn = ShLocnDescr0011
            imagelocn = ShLocnDescr0012 ]

[ Object001
  nodetype: individual
  isa:      an Object with
            son = ShapeObject001
            objecttype = Shape ]

[ ShLocnDescr0011
  nodetype: individual
  isa:      a ShapeLocationDescriptor with
            locates = ShapeObject001
            coordsystem = CoordSystem0011
            centroid = (21 36'N, 106 47'E) ]

[ ShLocnDescr0012
  nodetype: individual
  isa:      a ShapeLocationDescriptor with
            locates = ShapeFeature001
            coordsystem = CoordSystem0012
            silhouette = PointSet001 ]

[ ShapeFeature001
  //similar to ShapeObject001 ]
```

Prototype nodes such as ShapeObject and Ship reside in the model, and must be provided by the system developer. Individual nodes such as Ship001 and ShLocnDescr0011 are generated by the system in response to a query; they reside in the sketchmap.

Some further prototype node structure for the example might be:

```
[ PointObject
  nodetype:      basic prototype
  isa:           an Object with
                //...similar to ShapeObject
  son:           OneOf {...}
                //all point objects
  worldlocation: a PointLocationDescriptor
  imagelocation: A PointLocationDescriptor
                //distinguishing slots for point objects ]

[ LinearObject
  //...similar to ShapeObject and
  PointObject ]
```

```
[ Feature
  //...similar to Object ]

[ ShapeFeature
  //...similar to ShapeObject ]

[ LocationDescriptor
  nodetype:      abstract prototype ]

[ ShapeLocationDescriptor
  nodetype:      specialization prototype
  isa:           a LocationDescriptor
  locates:       OneOf {(a ShapeObject),
                        (a ShapeFeature)}
  coordsystem:   a CoordinateSystem
  centroid:      a PointSet
                //allows for "fuzziness"
  orientation:   an AngleRange
                //...ditto
  silhouette:    a PointSet ]

...similarly for Point, Linear, and
AreaLocationDescriptors

[ CoordinateSystem
  nodetype:      abstract prototype
  units:         a LengthUnitSpecification
  scale:         a NumberRange
                //length units / system unit
  transforms:    SetOf {(a Coordinate Transform)
                        (a Coordinate System)),
                        ...} ]

[ CoordinateTransform
  //two coord systems and a matrix ]

[ PointSet
  //choice of representations ]

[ Point2D
  //a coord system and two numbers ]
```

### 3.4. INNATE PROCEDURAL KNOWLEDGE

The system will have some innate procedural capabilities to augment the assertional and pictorial knowledge of the model. As these capabilities expand, the system will become more and more autonomous. Our immediate goals are to automate the selection and application of procedures and some inferencing about object locations.

#### 3.4.1. DESCRIBED PROCEDURES

At the highest (strategic) level, control is embedded in the form of user-written programs. However, the system will have a powerful procedural substructure that should facilitate the writing of these programs (Figure 2).

An executive procedure may be attached to a model node. The executive takes as partial input an incompletely-specified data object and returns a more-completely specified one. As previously described, data objects of a given



type have a standard form throughout the system. Thus one deals with executives at the level of the "operations" of KRL, i.e., what is to be done. The executives are responsible for how to do it; they must select and run procedures. To make these decisions, the executives have access to descriptions of available resources, the desired accuracy or immediacy of the result, and the present state of the model, sketchmap, and image structures.

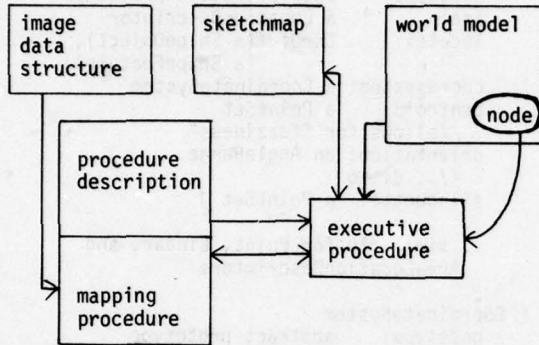


Figure 2

The executives must also be able to find out about the characteristics of procedures they are to use. For use by executives, the world-image mapping procedures (and others) must have associated descriptions containing:

- the slots in the data object which must be filled for the procedure to run;
- the slots the procedure can fill in;
- the cost of the procedure;
- the a priori reliability of the procedure.

With this scheme,

- executives can be written without considering the implementation details of mapping procedures in great depth,
- mapping procedures need not themselves determine an appropriate context for their application,
- descriptions allow a choice between methods (if several are available) based on capability, resource requirements, and a priori reliability,
- executives can select alternative procedures in the event of mapping procedure failure,
- if the mapping procedures can produce reliable a priori estimates of their success, the analytical results of [Bolles, Taylor] could be extended to select the procedure which most economically produces sufficiently precise answers.

### 3.4.2. INFERENCES ABOUT LOCATIONS

We expect our model occasionally to include maps which are more or less accurate metrical data structures. Such a map can be used to produce model structure such as:

```
PacificOcean isa <Area Object>;
OaklandDocks isa <Area Object>;
```

Here an area object has slots for such properties as a boundary, an area, etc.

The model will also have assertions such as:

```
DockedShips are ADJACENT and
PARALLEL to Docks or DockedShips;
Ships are IN Oceans;
Ship isa <Shape Object>;
```

To find docked ships using the above information, one is invited to search for a particular shape in the ocean, parallel to docks at some distance from them related to the width of a ship. We would like to give the system the capability of deriving this for itself, and thus being capable of intelligently instructing a ship-finder as to location and orientation of possible ships. We are developing the idea of putting constraints on an object's location via its location descriptor. The constraints are expressed in terms of objects whose locations may be known through maps, such as docks, or objects which may become known during image analysis, such as ships.

Geometric interpretation of constraint words such as IN and ADJACENT involves the construction of new location descriptors from old; intersection of area or boundaries is useful, as is the construction of parallel lines, bounding boxes, etc. Such construction of new or improved location descriptors will not be self-actuating, but the facility will be available to propagate and deduce knowledge about location whenever the time seems propitious. This sort of constraint satisfaction to reduce the range of parameters in a search seems useful in a variety of contexts, and it is being implemented as a built-in facility capable of using a variety of constraints on a variety of types of objects.

## REFERENCES

- Ballard, D.H., Brown, C.M., and Feldman, J.A.  
"Outline of a Goal-Directed Vision System,"  
submitted to IJCAI, 1978.
- Bobrow, D.S. and Winograd, T. "An Overview of  
KRL, a Knowledge Representation System,"  
Xerox PARC, CSL-76-4, July 1976.
- Bolles, R. "Verification Vision Within a Pro-  
grammable Assembly System," Stanford AI Memo  
AIM-275, December 1975.
- Feldman, J.A. and Rovner, P.D. "An Algol-Based  
Associative Language," CACM, Vol. 12, No. 8,  
August 1969, pp. 439-449.
- Freuder, E.D. Ph.D. Thesis, M.I.T. AI Lab, 1975.
- Lemmer, J. "Military Cartographic Products,"  
talk given at University of Rochester,  
September 1976.
- Maleson, J. and Rashid, R. "The Rochester Image  
Protocol," University of Rochester, Computer  
Science Department, Internal Memo, February  
1977.
- Minsky, M. "A Framework for Representing Know-  
ledge," in The Psychology of Computer  
Vision, Winston, P.H. (Ed.), McGraw-Hill,  
1975.
- Sproull, R.F. and Baudelaire, P. "Proposed  
'Array of Intensity' Samples Format," Xerox  
PARC Internal Document, May 1976.
- Taylor, R.H. "Generating AI Programs from High-  
Level Task Descriptions," Ph.D. Thesis,  
Stanford AI Lab, 1976.

## The Locus Model of Search and its Use in Image Interpretation

Steven M. Rubin & Raj Reddy

Department of Computer Science  
Carnegie-Mellon University, Pittsburgh, Pa. 15213  
March 26, 1977

### Introduction

The central problem in image understanding is the representation and use of all the available sources of knowledge in the interpretation and description of an image. The problem of representation is complicated by the diversity of sources of knowledge. Converting knowledge into effective algorithms in the presence of error and uncertainty further complicates the issue. In this paper we present a specific framework for representation and use of knowledge which appears to be both sufficient and efficient for a wide variety of image interpretation tasks.

The framework for image interpretation presented here is based on the Locus model successfully used in speech understanding research (Lowerre and Reddy, 1977). The Locus model is a non-backtracking, non-iterative, deterministic search technique in which a beam of near-miss alternatives around the best path are extended to determine the near-optimal description of the image.

In the following sections we will outline the structure of the model and discuss the relationship of the present approach to earlier attempts at image interpretation. A complete version of this paper, including a detailed example has been submitted to IJCAI-77 and can be obtained by writing to the authors. A detailed description of the model as applied to image interpretation task will be given in Rubin (1977). A more complete discussion of the strengths and limitations of the model and its relationship to the other approaches to knowledge representation and search are given in Reddy (1977).

### The Locus Model

The basic premise underlying the Locus model is that the problem of image interpretation can be viewed as a problem of search, and that given a specific knowledge representation paradigm and a specific signal-to-symbol transformation paradigm a highly efficient search can be used to obtain a globally optimal solution satisfying all the constraints of the world model.

The principal requirement of the Locus model is in the area of knowledge representation. Most approaches to image recognition assume the existence (and availability) of a world model in terms of some internal symbolic description. The world model usually consists of knowledge which defines the structure and relationship among objects that can occur in all the scenes that are interpretable by

the world model. By iteratively redefining higher level structures in terms of simpler objects one can generate a hierarchical network (or possibly a relational semantic network). The particular knowledge representation paradigm we have adopted in Locus is to attempt to represent all images that are admissible by the world model in terms of a graph structure whose nodes are Primitive Picture Elements (PPEs). A PPE is chosen so that all pixels belonging to a given PPE class share the same properties in the feature space (or signal space). Thus a PPE might sometimes represent an entire object as in the case of sky, river, or road, or represent a small subpart of an object such as a segment with similar textural properties. We therefore assume the existence of a set of PPEs which can be used to compose any image that is admissible by the world model. Further we assume that most, if not all, of the constraints about object structure, size, shape, location, and orientation are expressible in terms of the graph structure containing only the PPEs. It is obvious that this type of knowledge representation is likely to be expensive in terms of space for all except the most trivial problems but it appears to be what is needed for an efficient solution. Baker (1974) and Lowerre (1976) show how different types of knowledge and constraints can be combined into a single graph structure.

The second requirement of the Locus model is the availability of a signal-to-symbol transformation technique by means of which one can estimate the probability that a given PPE is present at (or around) a pixel location. This basically requires the availability of a pattern template for each PPE and a distance metric for matching the unknown signal with the PPE templates.

In the absence of any constraints, the optimal assignment of PPEs to pixels can be obtained by selecting the best PPE label in each pixel neighborhood. However, given the semantic, syntactic, structural, and segmental properties of scenes that are acceptable within a micro-world model, one wishes to choose that assignment of PPEs to pixels that is both globally optimal and consistent with the world model.

Given a PPE graph structure representation of the world model and a signal-to-symbol transformation technique, the problem of interpreting an unknown image can be viewed as finding the optimal path through the graph, i.e., finding a sequence of PPEs which best describe each of the pixel neighborhoods of the unknown image, subject to constraints defined by the knowledge sources represented by the graph.



Before we can define the search strategy for finding the optimal path we need to define the term path probability in a PPE network. Path probability is defined incrementally in terms of the nodes it traverses and uses three pieces of information to calculate a probability: the statistical match of the signal to the symbol; the probabilities of previous network nodes; and the transition probabilities of arriving from those previous network nodes. Formally:

$$P_{i,j} = A_{i,j} \times \text{AVERAGE} \left[ \max_k (P_{k,j+\Delta(d)} \times T_{k,i,d}) \right] \quad (1)$$

where  $P_{i,j}$  is the probability of being in network state  $i$  at position  $j$  of the sensed data;  $A_{i,j}$  is the statistical match of the PPE symbol represented by state  $i$  to the signal at position  $j$ ;  $\Delta(d)$  is the state adjacency function which offsets the current state ( $j$ ) to the previous state ( $j+\Delta(d)$ ) in direction  $d$ ; and  $T_{k,i,d}$  is the transition probability of traveling from state  $k$  to state  $i$  in direction  $d$ . For image processing, the position ( $j$ ) is an  $(x, y)$  vector. Note that the maximum  $k$  in the above equation is saved as the best previous state. This is how each node identifies the best path to take during the back-trace. Note also that the probability values are not needed during the back-trace: they accumulate on the forward pass only. The back-pointers are calculated on the forward pass using the probability information, so they reflect all node transitions to that point. The back-trace uses only the best previous node for each state as it quickly steps through the network and selects a path. No search is performed in this pass: it is pure look-up.

Finding the optimal path through the graph is a classical search problem in Artificial Intelligence with many possible alternative search strategies (Nilsson, 1971). In this paper we propose and use yet another search strategy called Locus which appears to be particularly effective in perceptual problem solving. Locus is a beam-search heuristic in which all except a beam of near-miss alternatives around the best path are pruned from the search tree at each pixel (or segmental) decision point, thus containing the exponential growth without requiring backtracking and non-deterministic search.

The Locus search proceeds as follows: 1) a forward pass calculates path probabilities and inter-node connections, and 2) a back-trace uses the inter-node connections to determine the components of the optimal network path. As the forward pass search progresses through the network, unpromising alternatives are pruned and the interconnections along the beam are saved until the end of the network is reached. At this point, a back-trace of the connections is made to select a path through the network. Note that this path is expected to lie in the beam that was carved out by the forward pass. By delaying the decision making process until all of the network nodes have been examined, Locus obtains the globally optimal path through the network. This is because the calculation of a node's likelihood hinges on all previous nodes that led up to it. Thus, during the back-trace, each node decision is guaranteed against degeneration because it's likelihood is supported by all nodes before it. This means that the selection of an object label in one corner of the scene can affect the labeling in the opposite corner. Consideration of

all of the near-miss alternatives removes the need for backtracking, and thus removes the problem of whether to search by depth or breadth.

## Discussion

The model presented in this paper has been used to interpret Ohlander's city scene, demonstrating the initial validity and usefulness of the model. We plan to use the model to interpret arbitrary views of downtown Pittsburgh (a 3-D world), and different satellite views of the Washington, D.C. area (a 2-D world). Representation of the knowledge about 3-D and 2-D world models in terms PPE graph structure requires the development of several preprocessing programs (the PPE graph for Ohlander's city scene was generated manually). In this section we will discuss the relationship of this model to other approaches in image recognition research, and our present views of the strengths and limitations of this approach.

The graph structure representation proposed here is a natural outgrowth of work in languages (Aho and Ullman, 1972) and syntax directed pattern recognition (Narasimhan, 1966; Clowes, 1969; and Fu, 1976). The approach presented in this paper principally differs from the above in how the network representation is to be used. It rejects the notion that image recognition is best viewed as a problem in parsing. Given the error and uncertainty associated with the decisions, the problem tends to be not one of deciding whether a given pattern is parsable but rather one of search, i.e., deciding which of the many acceptable alternative parses represents the optimal choice.

The view that the problem of image recognition is one of constraint satisfying search has been gaining increasing acceptance (Waltz, 1975; Tannenbaum and Barrow, 1976; Hummel, Zucker and Rosenfeld, 1976). This paper also subscribes to this viewpoint and differs mainly from the other efforts in the representation of constraints and the method of search.

The realization that one needs to introduce some measure of the degree of uncertainty into the interpretation process is reflected in the papers by Fischler and Elschlager (1973), Feldman and Yakimovsky (1975), and the probabilistic relaxation methods under development at SRI and Maryland. The method proposed here is able to handle search in the presence of error and uncertainty in a natural and straightforward manner provided all knowledge and constraints are represented in terms of a PPE graph structure.

Constraint satisfying search in the presence of uncertainty is also a central problem in other areas of AI, in particular in speech understanding systems research. Several techniques developed for use in the speech area such as representation of knowledge sources as cooperating independent processes (Reddy et al, 1973; Lesser et al., 1975; Erman et al., 1977), island driven search (Erman et al., 1977; Woods et al., 1977), and network representations of knowledge (Baker, 1975; Lowerre, 1976) also appear to be relevant to other knowledge based systems research, including vision. The Locus model presented here was first developed for use in the Harpy connected speech

recognition system. Though the basic ideas remain the same, the model had to be revised substantially to make it useful in image recognition.

The best-first search given by the A\* algorithm (Nilsson, 1971) and the breadth-first graph search of the dynamic programming algorithms (Bellman, 1962) provide alternative approaches to optimal graph search problems. The beam-search technique of the Locus model provides a minimal effort near-optimal solution and appears to be effective in cases where the evaluation function is a function of an external signal source and where a large number of decisions are related to each other in that they are all attempting to provide alternative interpretations of the same signal segment.

A significant feature of the Locus model of search is its linearity. Because Locus prunes all but a narrow beam of alternatives, its search time is linear with respect to the size of the input signal and is essentially independent of the symbol space size. Thus, Locus searching controls the combinatorial explosion that occurs in most graph searching techniques. Note, however, that the size of the beam expands and contracts during the search as the connectivity between symbols in the graph increases and with the degree of uncertainty of the decisions.

The order of search in Locus is a subtle issue that appears to be a problem but really isn't. When using Locus in speech understanding, there is one independent dimension of time which can be used to order the search. In image processing with static pictures, there are two dimensions, so a raster scan is used. This might appear to cause continuity problems especially at the end of a scan line. However, Locus requires only the local context for a point and it propagates the global context regardless of search order. Thus, any search pattern can be used as long as it is reversed on the back-trace. Note also that the raster scan has the advantage of allowing the use of context for horizontally, vertically, and diagonally adjacent states.

A main concern with the finite state networks is that not all relational constraints are easily representable within that framework. We have not found this to be a problem in the 3-D and 2-D worlds we have considered so far, although the representations tend to be expensive in space (memory) required. We expect to use a post-pass to apply constraints that are not easily incorporated into the network.

## Conclusion

This paper provides a framework for knowledge representation and search for image recognition tasks, leading to an easily implementable total systems framework within which one can explore the relative merits of different types of knowledge. One still has to decide what knowledge is available, how to acquire and define it, how to select an adequate set of primitive picture elements (PPE) for a given task, and how to match symbols (PPEs) to the signal. However, each of these subtasks look much more manageable to us than the original image interpretation task.

## References

- Aho, A. V. and J. D. Ullman (1972). The Theory of Parsing, Translation, and Compiling, Prentice-Hall, Englewood Cliffs, N. J.
- Baker, J. K. (1975). "The DRAGON system--An overview", IEEE Trans. Acoust. Speech, Signal Processing, vol. ASSP-23, 24-29, Feb.
- Bellman, R. and Dreyfus, S. (1962). Applied Dynamic Programming, Princeton Univ. Press, Princeton, N.J.
- Clowes, M. C. (1976). "Pictorial relationships - A Syntactic Approach", in Machine Intelligence IV (Meltzer and Michie, eds.), American Elsevier, New York.
- Erman, L. D., et al. (1977). The Hearsay-II System, (in preparation).
- Feldman, J. A. and Yakimovsky, Y. (1975). "Decision Theory and Artificial Intelligence: I. A Semantics Based Region Analyzer," Artificial Intelligence, vol 5, pp. 349-371.
- Fischler, M. A. and Eschlager, R. A. (1973). "The Representation and Matching of Pictorial Structures," IEEE Transactions on Computers, January.
- Fu, K. S. (1976). "Syntactic (Linguistic) Pattern Recognition," in Digital Pattern Recognition (Fu, ed.), Springer Verlag, New York.
- Hummel, R. A., Zucker, S. W. and Rosenfeld, A. (1976). "Scene labelling by relaxation operations," IEEE Trans. Syst. Man, Cybern.
- Lesser, V. R., Fennell, R. D., Erman, L. D., and Reddy, D. R. (1975). "Organization of the Hearsay-II Speech Understanding System," IEEE Trans. Hum. Factors Electron., vol. ASSP-23, 11-23.
- Lowerre, B. T. (1976). "The HARPY Speech Recognition System," (Ph.D. Thesis, Carnegie-Mellon University), Tech. Report, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA.
- Lowerre, B. T., and Reddy, D. R. (1977). "Representation and Search in the Harpy Connected Speech Recognition System," paper submitted to IJCAI-77.
- Narasimhan, R. (1966). "Syntax-Directed Interpretation of Classes of Pictures," Communications of the ACM, vol. 9, no. 3, March.
- Nilsson, N. (1971). Problem Solving Methods in Artificial Intelligence, McGraw Hill.
- Ohlander, R. B. (1975). "Analysis of Natural Scenes," (Ph.D. Thesis, Carnegie-Mellon University), Tech. Report, Computer Science Department, Carnegie-Mellon University.
- Reddy, D. R. (1977). "Aspects of Representation and Search in Perceptual Problem Solving," (in preparation).
- Reddy, D. R., Erman, L. D., and Neely, R. B. (1973). "A model and a system for machine recognition of speech," IEEE Trans. Audio Electroacoust., vol. AU-21, 229-238, June.
- Rubin, S. M. (1977). "The ARGOS Image Understanding System," (Ph.D. Thesis, Dept. of Computer Science, Carnegie-Mellon University), (in preparation).
- Tannenbaum, J. M. and Barrow, H. G. (1976). "Experiments in Interpretation-Guided Segmentation," Technical Note 123, Stanford Research Institute, Menlo Park, CA.
- Waltz, D. (1975). Understanding Line Drawings with Shadows, in The Psychology of Computer Vision, (P. Winston, Ed.), MIT Press, Cambridge, MA.
- Woods, W. W., et al. (1977). "Final Report on Speech Understanding Systems," Bolt, Beranek and Newman Inc., Cambridge, MA.



## Cooperative Computation of Stereo Disparity

D. Marr and T. Poggio

The Artificial Intelligence Laboratory, Massachusetts Institute of Technology,  
545 Technology Square, Cambridge, Mass. 02139, U. S. A.

### Abstract

The extraction of stereo disparity information from two images depends upon establishing a correspondence between them. This article analyzes the nature of the correspondence computation, and derives a cooperative algorithm that implements it. We show that this algorithm successfully extracts information from random-dot stereograms, and its implications for the psychophysics and neurophysiology of the visual system are briefly discussed.

### Introduction

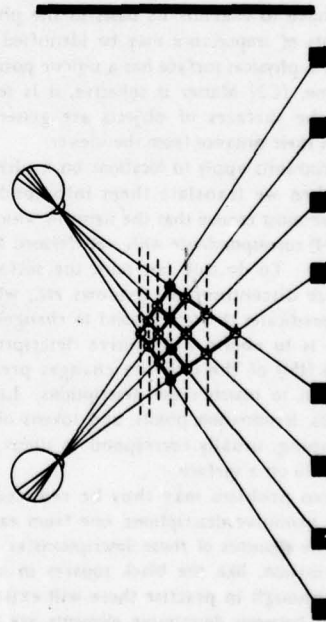
Perhaps one of the most striking differences between a brain and today's computers is the amount of wiring. In a digital computer, the ratio of connexions to components is about three, whereas for the mammalian cortex it lies between 10 and 10,000 (1).

Although this fact points to a clear structural difference between the two, it is important to realise that this distinction is not fundamental to the nature of the information processing that each accomplishes, merely to the particulars of how it does it. In Chomsky's terms (2), it affects theories of performance but not theories of competence, because the nature of a computation that is carried out by a machine or a nervous system depends only on the problem to be solved, not on the available hardware (3). Nevertheless one can expect a nervous system and a digital computer to use different types of algorithm, even when performing the same underlying computation. Algorithms with a parallel structure, requiring many simultaneous local operations on large data arrays, are expensive for today's computers but probably well-suited to the highly interactive organization of nervous systems.

The class of parallel algorithms includes an interesting and not precisely definable subclass which we may call *cooperative algorithms* (3). Such algorithms operate on many "input" elements and reach a global organisation *via* local, interactive constraints. The term "cooperative" refers to the way in which local operations appear to cooperate in forming global order in a well-regulated manner. Cooperative phenomena are well-known in physics (4, 5), and it has recently been proposed that they may play an important role in biological systems as well (4, 6, 7, 8, 9, 10). One of the earliest suggestions along these lines is due to Julesz (11), who maintains that stereoscopic fusion is a cooperative process. His spring and dipoles model represents a suggestive metaphor for this idea. Besides its biological relevance, the extraction of stereoscopic information is an important and yet unsolved problem in visual information

processing (12). For this reason -- and also as a case-in-point -- it seems interesting to describe a cooperative algorithm for this computation.

In this article, we shall (a) analyse the computational structure of the stereo-disparity problem, stating the goal of the computation and characterising the associated local constraints; (b) describe a cooperative algorithm that implements this computation; and (c) exhibit its performance on random-dot stereograms. Although the problem addressed here is not directly related to the question of how the brain extracts disparity information, we shall briefly mention some questions and implications for psychophysics and neurophysiology.



1. There is ambiguity in the correspondence between the two retinal projections. In this figure, each of the four points in one eye's view could match any of the four projections in the other eye's view. Of the 16 possible matchings only four are correct (filled circles), while the remaining 12 are "false targets" (open circles). It is assumed here that the targets (filled squares) correspond to "matchable" descriptive elements obtained from the left and right images. Without further constraints based on global considerations, such ambiguities cannot be resolved. Redrawn after Julesz (ref. 12 figure 4.5-1).

### Computational Structure of the Stereo-disparity Problem

Because of the way our eyes are positioned and controlled, our brains usually receive similar images of a scene taken from two nearby points at the same horizontal level. If two objects are separated in depth from the viewer, the relative positions of their images will differ in the two eyes. Our brains are capable of measuring this disparity, and using it to estimate depth.

Three steps are involved in measuring stereo disparity: (S1) a particular location on a surface in the scene must be selected from one image; (S2) that same location must be identified in the other image; and (S3) the disparity in the two corresponding image points must be measured.

If one could identify a location beyond doubt in the two images, for example by illuminating it with a spot of light, steps S1 and S2 could be avoided and the problem would be easy. In practise one cannot do this (see figure 1), and the difficult part of the computation is solving the correspondence problem. Julesz found that we are able to interpret random-dot stereograms, which are stereo pairs that consist of random dots when viewed monocularly, but which fuse when viewed stereoscopically to yield patterns separated in depth. This might be thought surprising because when one tries to set up a correspondence between two arrays of random dots, false targets arise in profusion (see figure 1). Yet we are able to determine the correct correspondence. We need no other cues.

In order to formulate the correspondence computation precisely, we have to examine its basis in the physical world. Two constraints of importance may be identified (13): (C1) A given point on a physical surface has a unique position in space at any one time; (C2) Matter is cohesive, it is separated into objects, and the surfaces of objects are generally smooth compared with their distance from the viewer.

These constraints apply to locations on a physical surface. Therefore when we translate them into conditions on a computation we must ensure that the items to which they apply there are in (1-1) correspondence with well-defined locations on a physical surface. To do this one must use surface markings, normal surface discontinuities, shadows *etc.*, which in turn means using predicates that correspond to changes in intensity. One solution is to obtain a primitive description (like the primal sketch (15)) of the intensity changes present in each image, and then to match these descriptions. Line and edge segments, blobs, termination points, and tokens obtained from these by grouping, usually correspond to items that have a physical existence on a surface.

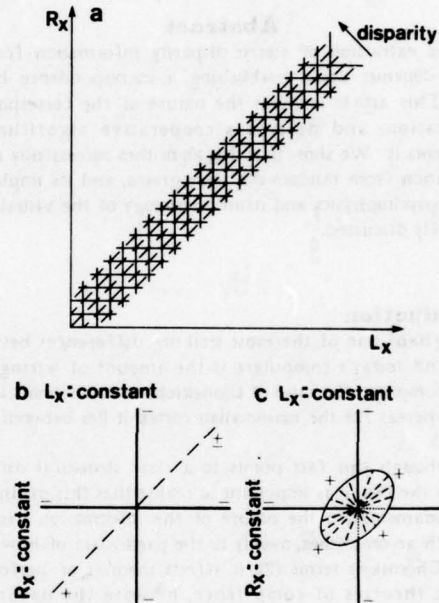
The stereo problem may thus be reduced to that of matching two primitive descriptions, one from each eye. One can think of the elements of these descriptions as carrying only position information, like the black squares in a random-dot stereogram, although in practise there will exist rules about which matches between descriptive elements are possible, and which are not. The two physical constraints C1 and C2 can now be translated into two rules for how the left and right descriptions are combined:

(R1) *Uniqueness.* Each item from each image may be assigned at most one disparity value. This condition relies on the assumption that an item corresponds to something that has a unique physical position.

(R2) *Continuity.* Disparity varies smoothly almost everywhere. This condition is a consequence of the cohesiveness of matter, and it states that only a small fraction of the area of an image

is composed of boundaries that are discontinuous in depth.

It is important to stress that in real life, R1 cannot be applied simply to grey-level points in an image. The simplest counter-example is that of a goldfish swimming in a bowl, because many points in the image receive contributions from the bowl and from the goldfish. Here, and in general, a grey-level point is in only implicit correspondence with a physical location, and it is therefore impossible to ensure that grey-level points in the two images correspond to exactly the same physical position. Sharp changes in intensity are usually due either to the goldfish, or to the bowl, or to a reflexion, and therefore define a single physical position precisely.



2. Figure 2a shows the explicit structure of the two rules R1 and R2 for the case of a one-dimensional image, and it also represents the structure of a network for implementing the algorithm described by equation 2. Solid lines represent "inhibitory" interactions, and dotted lines represent "excitatory" ones. 2b gives the local structure at each node of the network 2a. This algorithm may be extended to two-dimensional images, in which case each node in the corresponding network has the local structure shown in 2c. Such a network was used to solve the stereograms exhibited in figures 3 - 6.

### A Cooperative Algorithm

By constructing an explicit representation of the two rules, we can derive a cooperative algorithm for the computation. Figure 2a exhibits their geometry in the simple case of a one-dimensional image.  $L_x$  and  $L_y$  represent the positions of descriptive elements on the left and right images. The thick vertical and horizontal lines represent lines of sight from the left and right eyes, and their intersection points correspond to possible disparity values. The dotted diagonal lines connect points of constant disparity.

The uniqueness rule R1 states that only one disparity value



may be assigned to each descriptive element. If we now think of the lines in figure 2a as a network, with a node at each intersection, this means that only one node may be switched on along each horizontal or vertical line.

The continuity rule *R2* states that disparity values vary smoothly almost everywhere. That is, solutions tend to spread along the dotted diagonals.

If we now place a "cell" at each node (figure 2b), and connect it so that it inhibits cells along the thick lines in the figure, and excites cells along the dotted lines, then provided the parameters are appropriate the stable states of such a network will be precisely those in which the two rules are obeyed. It remains only to show that such a network will converge to a stable state, and we were able to carry out a combinatorial analysis (as in refs. 9 & 15) which established its convergence for random-dot stereograms (16).

This idea may be extended to two-dimensional images simply by making the local excitatory neighbourhood two-dimensional. The structure of each node in the network for two-dimensional images is shown in figure 2c.

A simple form of the resulting algorithm (3) is given by the following set of difference equations:

$$C^{(n+1)} = \sigma\{\Xi(C^{(n)}) + C^{(0)}\} \quad (1)$$

$$C_{xyd}^{(n+1)} = \sigma \left\{ \sum_{x',y',d' \in \mathcal{N}_{exc}(xyd)} C_{x'y'd'}^{(n)} - \epsilon \sum_{x',y',d' \in \mathcal{N}_{inh}(xyd)} C_{x'y'd'}^{(n)} + C_{xyd}^{(0)} \right\} \quad (2)$$

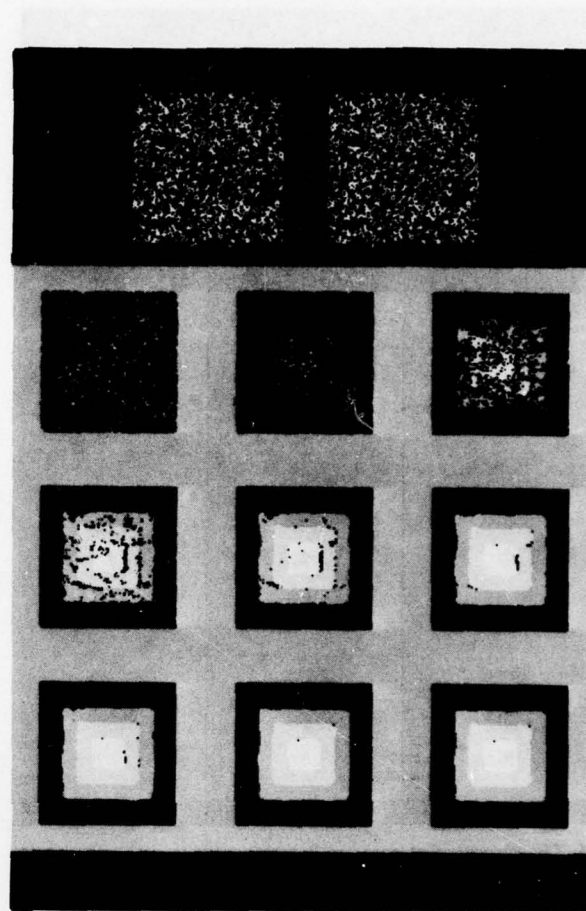
where  $C_{xyd}^{(n)}$  represents the state of the node or cell at position  $(x, y)$  with disparity  $d$  at iteration  $n$ ,  $\Xi$  is the linear operator that embeds the local constraints ( $\mathcal{S}$  and  $\mathcal{O}$  are the circular and thick line neighborhoods of the cell  $xyd$  in figure 2c), and  $\epsilon$  is the "inhibition" constant.  $\sigma$  is a sigmoid function with range [0, 1]. The state  $C_{xyd}^{(n+1)}$  of the corresponding node at time  $(n+1)$  is thus determined by a nonlinear operator on the output of a linear transformation of the states of neighbouring cells at time  $n$ .

The desired final state of the computation is clearly a fixed point of this algorithm, and moreover any state that is inconsistent with the two rules is not a stable fixed point. Our combinatorial analysis of this algorithm shows that, when  $\sigma$  is a simple threshold function, the process converges for a rather wide range of parameter values (16). The specific form of the operator is apparently not very critical.

Non-iterative local operations cannot solve the stereo problem in a satisfactory way (11). Recurrence and non-linearity are necessary to create a truly cooperative algorithm that cannot be decomposed into the superposition of local operations (17). General results concerning such algorithms seem to be rather difficult to obtain, although we believe that one can usually establish convergence in probability for specific forms of them.

### Examples of Applying the Algorithm

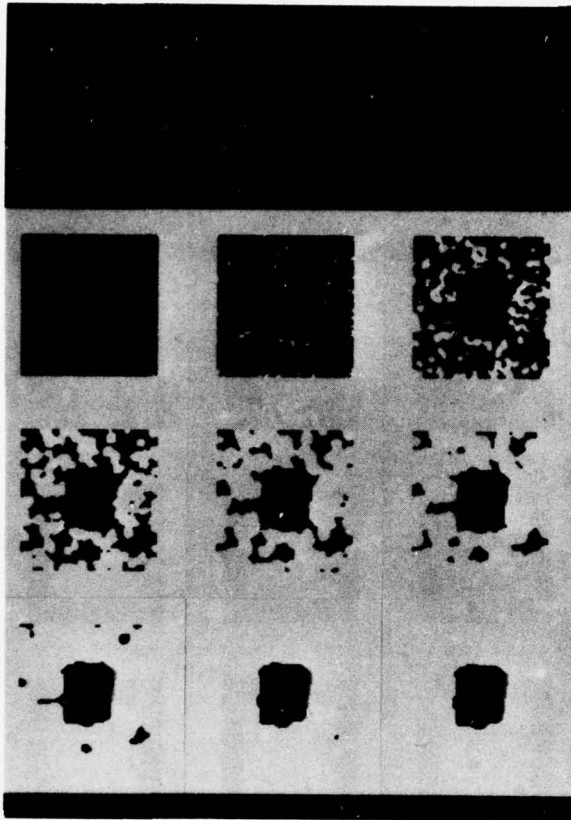
Random-dot stereograms offer an ideal input for testing the performance of the algorithm, since they enable one to bypass the costly and delicate process of transforming the intensity array received by each eye into a primitive description (14). When we ourselves view a random-dot stereogram, we probably compute a description couched in terms of edges rather than squares, whereas the inputs to our algorithm are the positions of the black squares. Figures 3, 4, 5 and 6 show some examples in which the iterative algorithm successfully solves the correspondence problem, thus allowing disparity values to be



3. This and the following figures show the results of applying the algorithm defined by equation 2 to two random-dot stereograms. The initial state of the network  $C$  is defined by the input such that a node takes the value 1 if it occurs at the intersection of a 1 in the left and right eyes (see figure 2), and it has value 0 otherwise. The network iterates on this initial state, and the parameters used here, as suggested by the combinatorial analysis, were

$\theta = 3.0$ ,  $\epsilon = 2.0$  and  $M = 5$ , where  $\theta$  is the threshold and  $M$  is the diameter of the "excitatory" neighborhood illustrated in figure 2c. The stereograms themselves are labelled LEFT and RIGHT, the initial state of the network as 0, and the state after  $n$  iterations is marked as such. To understand how the figures represent states of the network, imagine looking at it from above. The different disparity layers in the network lie in parallel planes spread out horizontally, so that the viewer is looking down through them. In each plane, some nodes are on and some are off. Each of the seven layers in the network has been assigned a different gray level, so that a node that is switched on in the top layer (corresponding to a disparity of +3 pixels) contributes a dark point to the image, and one that is switched on in the lowest layer (disparity = -3) contributes a lighter point. Initially (iteration 0) the network is disorganized, but in the final state, stable order has been achieved (iteration 14), and the inverted wedding-cake structure has been found. The density of this stereogram is 50%.





4. The algorithm of equation 2, with parameter values given in the legend to figure 3, is capable of solving random-dot stereograms with densities from 50% down to less than 10%. For this and smaller densities, the algorithm converges increasingly slowly. If a simple homeostatic mechanism is allowed to control the threshold  $\theta$  as a function of the average activity (number of "on" cells) at each iteration (compare ref. 15), the algorithm can solve stereograms whose density is very low. In this example, the density is 5% and the central square has a disparity of +2 relative to the background. The algorithm "fills in" those areas where no dots are present, but it takes several more iterations to arrive near the solution than in cases where the density is 50%. When we look at a sparse stereogram, we perceive the shapes in it as cleaner than those found by the algorithm. This seems to be due to subjective contours that arise between dots that lie on shape boundaries.

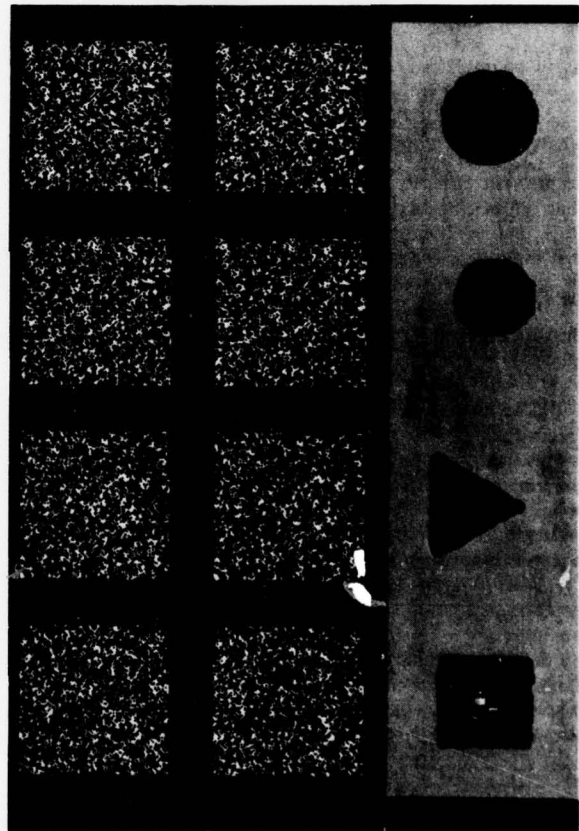
assigned to items in each image. Presently, its technical applications are limited only by the preprocessing problem.

This algorithm can of course be realised by various mechanisms, but parallel, recurrent, nonlinear interactions, both excitatory and inhibitory, seem the most natural. The difference equations set out above would then represent an approximation to the differential equations that describe the dynamics of the network.

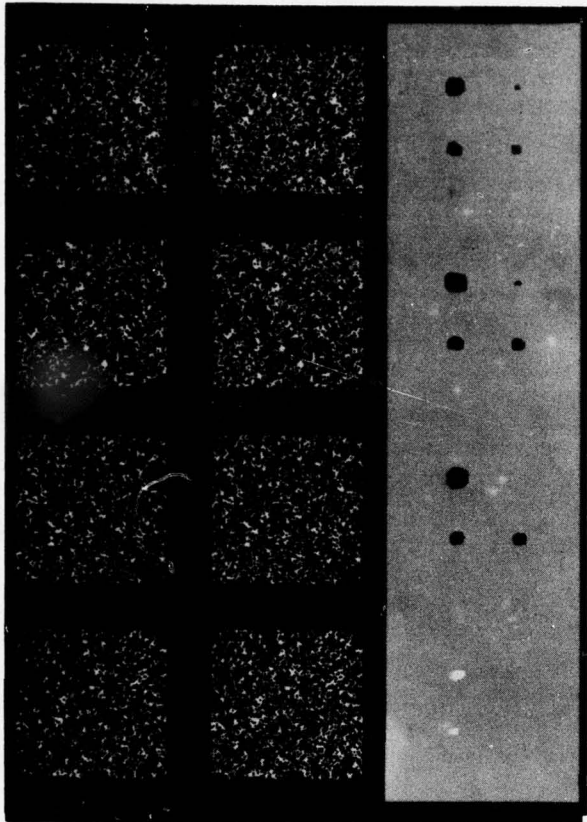
### Implications for Biology

We have hitherto refrained from discussing the biological problem of how stereopsis is achieved in the mammalian brain.

Our analyses of the computation, and of the cooperative algorithm that implements it, raise several precise questions for psychophysics and physiology. An important preliminary point concerns the relative importance of neural fusion and of eye-movements for stereopsis. The underlying question is, are there many disparity "layers" (as our algorithm requires), or are there just three "pools" (18) -- crossed, uncrossed and zero disparity. Most physiologists and psychologists seem to accept the existence of numerous, sharply tuned binocular "disparity detectors", whose peak sensitivities cover a wide range of disparity values (19, 20). We do not feel that the available evidence is decisive (21), but an answer is critical to the biological relevance of our analysis. If for example there were only three pools or layers with a narrow range of disparity sensitivities, the problem of false targets is virtually removed, but at the expense of having to pass the convergence plane of the eyes across a surface in order to achieve fusion. Psychophysical experiments are presently under way to gain some insight into this problem, but we believe that only physiology is capable of providing a clear-cut answer.



5. The disparity boundaries found by the algorithm do not depend on their shapes. In figures a, b and c we give examples of a circle, an octagon (notice how well the difference between them is preserved) and a triangle. The fourth example (d) shows a square in which the correlation is 100% at the boundary, but diminishes to 0% in the center. When one views this stereogram, the center appears to shimmer in a peculiar way. In the network, the center is unstable.



6. The width of the minimal resolvable area increases with disparity. In all four stereograms the pattern is the same, and consists of five circles with diameters 3, 5, 7, 9 and 13 dots. The disparity values exhibited here are +1, +2, +3 and +6, and for each pattern, we show the state of network after 10 iterations. As far as the network is concerned, the last pair (disparity +6) is uncorrelated, since only disparities from -3 to +3 are present in our implementation. After 10 iterations, information about the lack of correlation is preserved in the two largest areas.

If this preliminary question is settled in favour of a "multi-layer" cooperative algorithm, there are several obvious implications of the network (figure 2) at the physiological level: (a) the existence of many sharply tuned disparity units, that are rather insensitive to the nature of the descriptive element to which they may refer; (b) their organisation into disparity layers (or stripes or columns); (c) the presence of reciprocal excitation within each layer; (d) the presence of reciprocal inhibition between layers along the two lines of sight. Ideally, the inhibition should exhibit the characteristic "orthogonal" geometry of the thick lines in figure 2, but slight deviations may be permissible (16).

At the psychophysical level, several experiments (under stabilized image conditions) could provide critical evidence for or against the network: (a) results about the size of Panum's area and the number of disparity "layers"; (b) results about "pulling" effects in stereopsis (20); (c) results about the relationship between disparity and the minimum fusible pattern size (see fig. 6).

## Discussion

Our algorithm performs a computation that finds a correspondence function between two descriptions, subject to the two constraints of uniqueness and continuity. More generally, if one has a situation where "allowable" solutions are those that satisfy certain local constraints, a cooperative algorithm can often be constructed so as to find the "nearest" allowable state to an initial one. Provided that the constraints are local, use of a cooperative algorithm allows the representation of global order, to which the algorithm converges, to remain implicit in the network's structure.

The interesting difference between this stereo algorithm and standard correlation techniques is that one is not required to specify minimum or maximum correlation areas, to which the analysis is subsequently restricted. Previous attempts at implementing automatic stereocomparison through local correlation measurement have failed in part because no single neighbourhood size is always correct (12). The absence of a "characteristic scale" is one of the most interesting properties of this algorithm, and it is a central feature of several cooperative phenomena (22). We conjecture that the matching operation implemented by the algorithm represents in some sense a generalised form of correlation, subject to the *a priori* requirements imposed by the constraints. The idea is easily generalisable to different constraints and to other forms of equations (1) or (2), and it is technically quite appealing.

Cooperative algorithms may have many useful applications, (for example to best-match associative retrieval problems (15)), but their relevance to early processing of information by the brain remains an open question (23). Although a range of early visual processing problems might yield to a cooperative approach ("filling-in" phenomena, subjective contours (24), grouping, figural reinforcement, texture "fields", the correspondence problem for motion), it is important to emphasize that in problems of biological information processing, the first important and difficult task is to formulate the underlying computation precisely (3). After that, one can study good algorithms for it. In any case, we feel that an experimental answer to the question of whether depth perception is actually a cooperative process is a critical prerequisite to further attempts at analysing other perceptual processes in terms of similar algorithms.

## References and Notes

1. D. A. Sholl, *The Organisation of the Cerebral Cortex* (Methuen, London, 1956). The comparison depends of course on what is meant by a component. We refer here to the level of a gate and of a neuron, respectively.
2. A. N. Chomsky, *Aspects of the Theory of Syntax*. (M.I.T. Press, Cambridge Mass., 1965).
3. D. Marr and T. Poggio, in *The Visual Field: Psychophysics and Neurophysiology. Neurosciences Research Program Bulletin*, E. Poeppel et al., Eds. (in the press). Also available as M. I. T. A. I. Lab. Memo 357.
4. H. Haken, Ed. *Synergetics-Cooperative Phenomena in Multicomponent Systems*. (Teubner, Stuttgart, 1973).
5. H. Haken, *Rev. of Mod. Phys.* 47, 67 (1975).
6. J. D. Cowan, The problem of organismic reliability, in *Progress in Brain Research*, N. Wiener & J. P. Schade Eds., vol. 17. Amsterdam, Elsevier (1965).

7. H. R. Wilson and J. D. Cowan, *Kybernetik* 13, 55 (1973).
8. M. Eigen, *Naturwissenschaften* 58, 465 (1971).
9. P. H. Richter, *Die Phenomenologie der Immune Antwort*. (Contribution to a competition of the Bavarian Academy of Science, Max-Planck-Institut für Biophys. Chemie, 1974).
10. A. Gierer and H. Meinhardt, *Kybernetik* 12, 30 (1972).
11. B. Julesz, *Foundations of Cyclopean Perception* (Univ. of Chicago Press, Chicago, 1971).
12. K. Mori, M. Kidode and H. Asada, *Comp. Graphics and Image Processing* 2, 393 (1973).
13. D. Marr, *M. I. T. A. I. Lab. Memo* 327 (1974).
14. D. Marr, Early processing of visual information, *Phil. Trans. Roy. Soc. B*, (in the press).
15. D. Marr, *Phil. Trans. Roy. Soc. B* 252, 23 (1971). See especially section 3.1.2.
16. D. Marr and T. Poggio, in preparation.
17. T. Poggio and W. Reichardt, Visual control of orientation behaviour of the fly, part II. *Quarterly Rev. in Biophysics*, (in the press).
18. W. Richards, *J. Opt. Soc. Amer.* 62, 410 (1971).
19. H. B. Barlow, C. Blakemore and J. D. Pettigrew, *J. Physiol. (Lond.)* 193, 327 (1967); J. D. Pettigrew, T. Nikara and P. O. Bishop, *Exp. Brain Res.* 6, 391 (1968); C. Blakemore, *J. Physiol. (Lond.)* 209, 155 (1970).
20. B. Julesz and J.-J. Chang, *Bio. Cybernetics* 22, 107 (1976).
21. D. H. Hubel and T. N. Wiesel, *Nature* 225, 41 (1970).
22. K. G. Wilson, *Rev. of Modern Physics* 47, 773 (1975).
23. Julesz (11), Cowan (6), and Wilson & Cowan (7) were the first to discuss explicitly the cooperative aspect of visual information processing. A large literature has recently been accumulating on possible cooperative processes in nervous systems, ranging from the "catastrophe" literature (E. C. Zeeman, *Scientific American* 234, 65, April 1976) to various attempts of more doubtful credibility. There has hitherto been no careful study of a cooperative algorithm in the context of a carefully defined computational problem (but see ref. 15), although algorithms that may be interpreted as cooperative were discussed, for instance, by P. Dev, *Int. J. Man-Machine Studies* 7, 511 (1975); and by A. Rosenfeld, R. A. Hummel and S. W. Zucker, *IEEE Trans. SMC-6*, 420 (1976). In particular neither Dev nor J. I. Nelson, *J. theor. Biol.* 49, 1 (1975) formulated the computational structure of the stereo-disparity problem. As a consequence, the resulting geometry of the inhibition between their disparity detectors does not correspond to ours (see figure 2c) and apparently fails to provide a satisfactory algorithm.
24. S. Ullmann, *M. I. T. A. I. Lab. Memo* 367 (1976).
25. We thank Whitman Richards for valuable discussions, Henry Lieberman for making it easy to create stereograms, and Karen Prendergast for preparing the figures. This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75-C-0643. T. P. acknowledges the support of the Max-Planck-Gesellschaft during his visit to M.I.T.



## PARAMETRIC CORRESPONDENCE AND CHAMFER MATCHING:

## TWO NEW TECHNIQUES FOR IMAGE MATCHING

H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, H.C. Wolf

Artificial Intelligence Center  
 Stanford Research Institute  
 Menlo Park, California 94025

## Abstract

Parametric correspondence is a technique for matching images to a three dimensional symbolic reference map. An analytic camera model is used to predict the location and appearance of landmarks in the image, generating a projection for an assumed viewpoint. Correspondence is achieved by adjusting the parameters of the camera model until the appearances of the landmarks optimally match a symbolic description extracted from the image.

The matching of image and map features is performed rapidly by a new technique, called "chamfer matching", that compares the shapes of two collections of shape fragments, at a cost proportional to linear dimension, rather than area. These two techniques permit the matching of spatially extensive features on the basis of shape, which reduces the risk of ambiguous matches and the dependence on viewing conditions inherent in conventional image-based correlation matching.

## Introduction

Many military tasks require the ability to put a sensed image into correspondence with a reference image or map. Examples include vehicle guidance (navigation and terminal homing), photo interpretation (change detection and monitoring), and cartography (map updating). The conventional approach is to determine a large number of points of correspondence by correlating small patches of the reference image with the sensed image. A polynomial interpolation is then used to estimate correspondence for arbitrary intermediate points [Bernstein]. This approach is computationally expensive and limited to cases where the reference and sensed images were obtained under similar viewing conditions. In particular, it cannot match images obtained from radically

different viewpoints, sensors, or seasonal or climatic conditions, and it cannot match images against symbolic maps.

Parametric correspondence matches images to a symbolic reference map, rather than a reference image. The map contains a compact three-dimensional representation of the shape of major landmarks, such as coastlines, buildings, and roads. An analytic camera model is used to predict the location and appearance of landmarks in the image, generating a projection for an assumed viewpoint. Correspondence is achieved by adjusting the parameters of the camera model (i.e. the assumed viewpoint) until the appearances of the landmarks optimally match a symbolic description extracted from the image.

The success of this approach requires the ability to rapidly match predicted and sensed appearances after each projection. The matching of image and map features is performed by a new technique, called "chamfer matching", that compares the shapes of two collections of curve fragments at a cost proportional to linear dimension, rather than area.

In principle, this approach should be superior, since it exploits more knowledge of the invariant three dimensional structure of the world and of the imaging process. At a practical level, this permits matching of spatially extensive features on the basis of shape, which reduces the risk of ambiguous matches and dependence on viewing conditions.

## Chamfer Matching

Point landmarks, such as intersections or promontories, are represented in the map with their associated three dimensional world coordinates. Linear landmarks, such as roads or coastlines, are represented as curve fragments with associated ordered lists of world coordinates. Volumetric structures, such as buildings or bridges, are represented as wire-frame models.

From a knowledge of the expected viewpoint, a prediction of the image can be made by projecting world coordinates into corresponding image coordinates, suppressing hidden lines. The problem in matching is to determine how well the predicted features correspond with image features, such as edges and lines.

The first step is to extract image features by applying edge and line operators or tracing boundaries. Edge fragment linking [Nevatia, Perkins] or relaxation enhancement [Zucker, Barrow] is optional. The net result is a feature array each element of which records whether or not a line fragment passes through it. This process preserves shape information and discards greyscale information, which is less invariant.

To correlate the extracted feature array directly with the predicted feature array would encounter several problems: The correlation peak for two identical curves is very sharp and therefore intolerant of slight misalignment or distortions [Andrus]: A sharply peaked correlation surface is an inappropriate optimization criterion because it provides little indication of closeness to the true match, nor of the proper direction in which to proceed: Computational cost is heavy with large feature arrays.

A more robust measure of similarity between the two sets of feature points is the sum of the distances between each predicted feature point and the nearest image point. This can be computed efficiently by transforming the image feature array into an array of numbers representing distance to the nearest image feature point. The similarity measure is then easily computed by stepping through the list of predicted features and simply summing the distance array values at the predicted locations. The distance values can be determined by a process known as "chamfering", in two passes through the image feature array [Munson, Rosenfeld]. Note that this determination is made only once, after image feature extraction.

Chamfer matching provides an efficient way of computing the integral distance (i.e. area), or integral squared distance, between two curve fragments, two commonly used measures of shape similarity.

#### Parametric Correspondence

Parametric correspondence puts an image into correspondence with a three dimensional reference map by determining the parameters of an analytic camera model (3 position and 3 orientation parameters).

The traditional method of calibrating the camera model takes place in two stages: first, a number of known landmarks are independently located in the image, and second, the camera parameters are computed from the pairs of corresponding world and image locations, by solving an over-constrained set of equations [Sobel, Quam, Hannah].

The failings of the traditional method stem from the first stage. The landmarks are found individually, using only very local context (e.g. a small patch of surrounding image) and with no mutual constraints. Thus local false matches commonly occur. The restriction to small features is mandated by the high cost of area correlation, and by the fact that large image features correlate poorly over small changes in viewpoint.

Parametric correspondence overcomes these failings by integrating the landmark-matching and camera-calibration stages. It operates by hill-climbing on the camera parameters. A transformation matrix is constructed for each set of parameters considered, and it is used to project landmark descriptions from the map onto the image at a particular translation, rotation, scale and perspective. A similarity score is computed with chamfer matching and used to update parameter values. Initial parameter values are estimated from navigational data.

Integrating the two stages allows the simultaneous matching of all landmarks in their correct spatial relationships. Viewpoint problems with extended features are avoided because features are precisely projected by the camera model prior to matching. Parametric correspondence has the same advantages as rubber-sheet template matching [Fischler, Widrow] in that it obtains the best embedding of a map in an image, but avoids the combinatorics of trying arbitrary distortions by only considering those corresponding to some possible viewpoint.

#### An Example

The following example illustrates the major concepts in chamfer matching and parametric correspondence. A sensed image (Figure 1) was input along with manually derived initial estimates of the camera parameters. A reference map of the coastline was obtained, using a digitizing tablet to encode coordinates of a set of 51 sample points on a USGS map. Elevations for the points were entered manually. Figure 2 is an orthographic projection of this three dimensional map.

A simple edge follower traced the high contrast boundary of the harbor, producing the edge picture shown in Figure 3. The chamfering algorithm was applied to this edge array to obtain a distance array. Figure 4 depicts this distance array; distance is encoded by brightness with maximum brightness corresponding to zero distance from an edge point.

Using the initial camera parameter estimates, the map was projected onto the sensed image (Figure 5). The average distance between projected points and the nearest edge point, as determined by chamfer matching, was 25.8 pixels.

A straightforward optimization algorithm adjusted the camera parameters, to minimize the average distance. Figures 6 and 7 show an intermediate state and the final state, in which the average distance has been reduced to 0.8 pixels. This result, obtained with 51 sample points, compares favorably with a 1.1 pixel average distance for 19 sample points obtained using conventional image chip correlation followed by camera calibration. The curves in Figure 8 characterize the local behavior of this minimum, showing how average distance varies with variation of each parameter from its optimal value.

#### Discussion

We have presented a scheme for establishing correspondence between an image and a reference map that integrates the processes of landmark matching and camera calibration. The potential advantages of this approach stem from 1) matching shape, rather than brightness, 2) matching spatially extensive features, rather than small patches of image, 3) matching simultaneously to all features, rather than searching the combinatorial space of alternative local matches, 4) using a compact three dimensional model, rather than many two dimensional templates.

Shape has proved to be much easier to model and predict than brightness. Shape is a relatively invariant geometric property whose appearance from arbitrary viewpoints can be precisely predicted by the camera model. This eliminates the need for multiple descriptions, corresponding to different viewing conditions, and overcomes difficulties of matching large features over small changes of viewpoint.

The ability to treat the entirety of the relevant portion of the reference map as a single extensive feature reduces significantly the risk of ambiguous

matches, and avoids the combinatorial complexity of finding the optimal embedding of multiple local features.

A number of obstacles have been encountered in reducing the above ideas to practice. The distance metric used in chamfer matching provides a smooth, monotonic measure near the correct correspondence, and nicely interpolates over gaps in curves. However, scores can be unreliable when image and reference are badly out of alignment. In particular, discrimination is poor in textured areas, aliasing can occur with parallel linear features, a single isolated image feature can support multiple reference features.

The main problem is that edge position is not a distinguishing feature, and consequently many alternative matches receive equal weight. One way of overcoming this problem, therefore, is to use more descriptive features: brightness discontinuities can be classified, for example, by orientation, by edge or line, and by local spatial context (texture versus isolated boundary). Each type of feature would be separately chamfered and map features would be matched in the appropriate array. Similarly, features at a much higher level could be used, such as promontory or bay, area features having particular internal textures or structures, and even specific landmarks, such as "the top of the Transamerica pyramid". Ideally, with a few highly differentiated features distributed widely over the image the parametric correspondence process would be able to home in directly on the solution regardless of initial conditions.

Another dimension for possible improvement is the chamfering process itself. Determining for each point of the array a weighted sum of distances to many features (e.g. a convolution with the feature array), instead of the distance to the nearest feature, would provide more immunity from isolated noise points. Alternatively, propagating the coordinates of the nearest point instead of merely the distance to it, it becomes possible to use characteristics of features, such as local slope or curvature, in evaluating the goodness of match. It also makes possible a more directed search, since corresponding pairs of points are now known, an improved set of parameter estimates can be analytically determined.

Chamfer matching and parametric correspondence are separable techniques. Conceptually, parametric correspondence can be performed by re-projecting image chips and evaluating the match with correlation. However, the cost of



projection and matching grows with the square of the template size: The cost for chamfer matching grows linearly with the number of feature points. Chamfer matching is an alternative to other shape matching techniques, such as chain-code correlation [Freeman], Fourier matching [Zahn], and graph matching [e.g. Davis]. Also, the smoothing obtained by transforming two edge arrays to distance arrays via chamfering can be used to improve the robustness of conventional area-based edge correlation.

Parametric correspondence, in its most general form, is a technique for matching two parametrically related representations of the same geometric structure. The representations can be two- or three-dimensional, iconic or symbolic; the parametric relation can be perspective projection, a simple similarity transformation, a polynomial warp, and so forth. This view is similar to rubber-sheet template matching as conceived by Fischler and Widrow [Fischler, Widrow]. The feasibility of the approach in any application, as Widrow points out, depends on efficient algorithms for "pattern stretching, hypothesis testing, and pattern memory", corresponding to our camera model, chamfer matching, and three dimensional map.

As an illustration of its versatility, the technique can be used with a known camera location to find a known object whose position and orientation are known only approximately. In this case, the object's position and orientation are the parameters; the object is translated and rotated until its projection best matches the image data. Such an application has a more iconic flavor, as advocated by Shepard [Shepard], and is more integrated than the traditional feature extraction and graph matching approach [Roberts, Falk and Grape].

As a final consideration, the approach is amenable to efficient hardware implementation. There already exists commercially available hardware for generating parametrically specified perspective views of wire frame models at video rates, complete with hidden line suppression. The chamfering process itself requires only two passes through an array by a local operator, and match scoring requires only summing table lookups in the resulting distance array.

#### Conclusion

Iconic matching techniques, such as correlation, are known for efficiency and precision obtained by exploiting all available pictorial information,

especially geometry. However, they are overly sensitive to changes in viewing conditions and cannot make use of non-pictorial information. Symbolic matching techniques, on the other hand, are more robust because they rely on invariant abstractions, but are less precise and less efficient in handling geometrical relationships. Their applicability in real scenes is limited by the difficulty of reliably extracting the invariant description. The techniques we have put forward offer a way of combining the best features of iconic and symbolic approaches.

#### References

- Andrus, J.F., Campbell, C.W. and Jayroe, R.R., "A Digital Image Registration Method Using Boundary Maps", IEEE Trans. Comp., Sept. 1975, p. 935-939.
- Barrow, H.G., "Interactive Aids for Cartography and Photo Interpretation", Interim Report of ARPA Project DAAG29-76-C-0057, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, Dec. 1976.
- Bernstein, R., "Digital Image Processing of Earth Observation Sensor Data", IBM Journal of Research and Development, Vol. 20, No. 1, 1976.
- Davis, L., "Shape Matching Using Relaxation Techniques", TR-480, Computer Science Dept., University of Maryland, Sept. 1976.
- Falk, G., "Interpretation of Imperfect Line Data as a Three Dimensional Scene," Artificial Intelligence, Vol. 3, 1972, p.101-144.
- Feder, J. and Freeman, H., "Segment Fitting of Curves in Pattern Analysis Using Chain Correlation", AD619525, March 1965.
- Fischler, M. and Elschlager, R., "The Representation and Matching of Pictorial Structures", IEEE Trans. Comp., no. 22, 1973, p.67-92.
- Grape, G.R., "Model Based (Intermediate Level) Computer Vision", Stanford AI Memo AIM-201, May 1973, Ph.D. Thesis.
- Hannah, M.J., "Computer Matching of Areas in Stereo Images", Stanford AI Memo AIM-239, July 1974, Ph.D. Thesis.

- Munson, J., Internal memo, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, Dec. 1973.
- Perkins, W.P., "Multi-level Vision Recognition System", Proc. Third IJCPR, Nov. 1976, p.739-744.
- Quam, L.H., "Computer Comparison of Pictures", Stanford AI Project Memo AIM-144, May 1971, Ph.D. Thesis.
- Roberts, L.G., "Machine Perception of Three-dimensional Objects", in: Tippet, J.T., et al. (Eds.), "Optical and Electro-Optical Information Processing", MIT Press, 1965, p.159-197.
- Rosenfeld, A. and Pflatz, J.L., "Distance Functions on Digital Pictures", Pattern Recognition, Vol. 1 No. 1, July 1968, p.33-62.
- Shepard, R.N. and Metzler, J., "Mental Rotation of Three-Dimensional Objects", Science, 1971, p.701-703.
- Sobel, I., "On Calibrating Computer Controlled Cameras for Perceiving 3-D Scenes", Artificial Intelligence, Vol. 5, 1974, p.185-198.
- Widrow, B., "The 'Rubber-Mask' Technique", Pattern Recognition, Vol. 5, 1973, p.175-211.
- Zahn, C. and Roskies, R., "Fourier Descriptors for Plane Closed Curves", IEEE Trans. Comp., no. 21, 1972, p.269-281.
- Zucker, S., Hummel, R. and Rosenfeld, A., "An Application of Relaxation Labeling to Curve and Line Enhancement", IEEE Trans. Comp., no. 25, 1976.



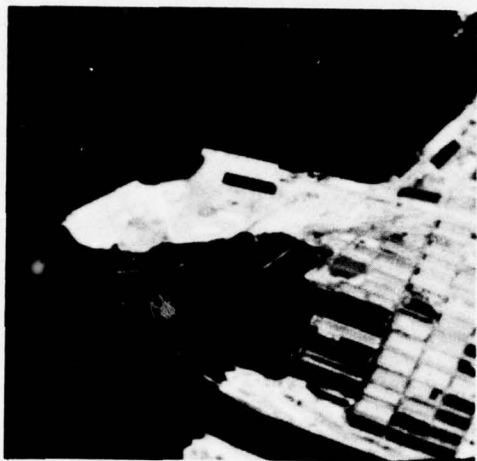


Figure 1. An aerial image of a section of coastline.

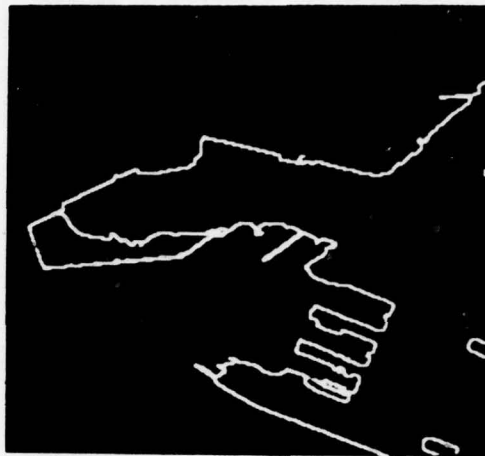


Figure 3. The traced boundary of the coastline.

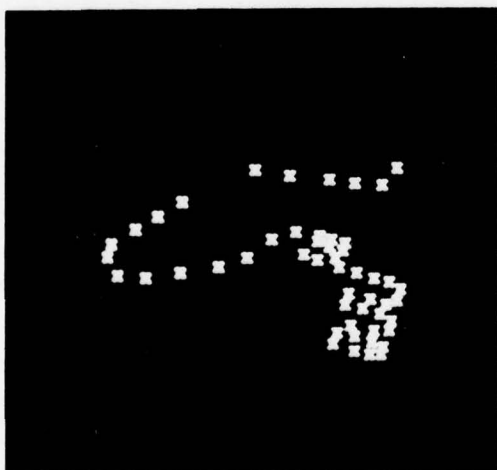


Figure 2. A set of sample points taken from a USGS map.

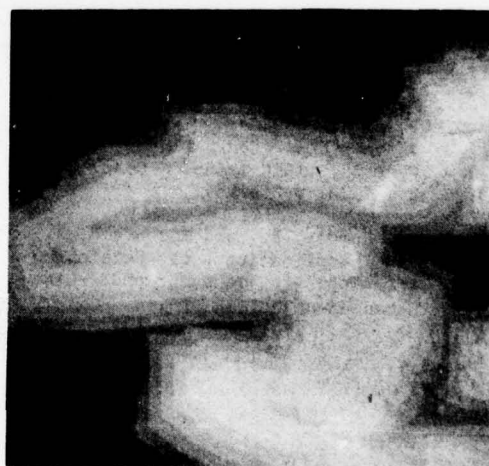


Figure 4. The distance array produced by chamfering the boundary.



Figure 5. Initial projection of map points onto the image.

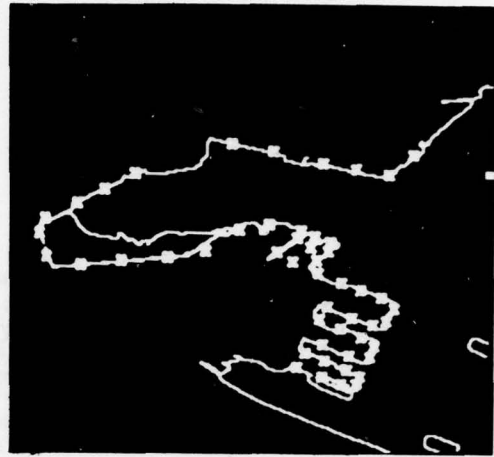


Figure 7. Projection of map points onto the image after optimization of camera parameters.



Figure 6. Projection of map points onto the image after some adjustment of camera parameters.

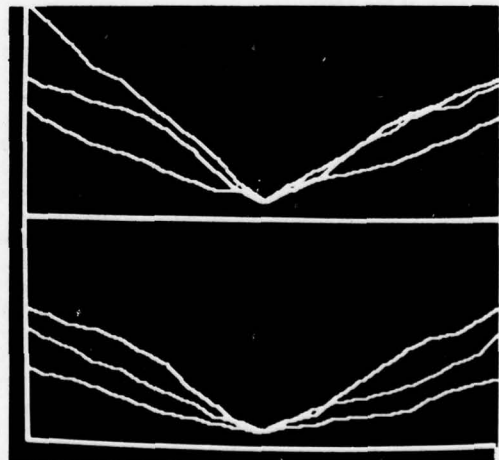


Figure 8. Behavior of average distance score with variation of the six camera parameters from their optimal values.

## SYMBOLIC IMAGE REGISTRATION AND CHANGE DETECTION

Keith Price and D. Raj Reddy

Department of Computer Science  
Carnegie-Mellon University  
Pittsburgh, Pa. 15213

### Summary

This paper describes research toward the development of symbolic registration and change detection techniques directed toward the problem of the comparison of pairs of different images of the same scene to generate descriptions of the changes in the scene. Unlike earlier work in the change analysis area, all the matching and change analysis is performed at a symbolic level rather than a signal level. To facilitate this symbolic analysis over a wide variety of images, advances in several other areas of image analysis are also required. These areas are: segmentation techniques to generate the basic units used in the symbolic analysis, feature analysis to generate the symbolic description of the regions and image, use of knowledge to guide the segmentation and symbolic registration procedures, and lastly change analysis itself. We applied this procedure on several diverse scenes (house, cityscape, satellite images, aerial images, and radar images), each of which included a task description and a predefined set of knowledge elements, and have shown how several different tasks can be performed with a general change analysis system.

### Summary of the Tasks

The scenes which we analyze (see Price, 1976) for a more complete description) are: a simple house scene, a cityscape scene, a LANDSAT (satellite) scene showing snow cover changes, a SLR (side looking radar) scene, an aerial rural scene, and an aerial urban or industrial scene. The first two scenes have three initial spectral inputs, the LANDSAT scene has four, and the other three have only one (intensity of radar signal or visible light). The tasks are: Perform simple symbolic registration for the house scene. Perform symbolic registration in a more textured scene with changes in the relative position of objects using the cityscape scene. Perform the analysis of a different spectral domain (radar) and symbolic registration with the SLR scene. Perform symbolic registration in the presence of rotations in the aerial rural scene. Perform symbolic registration and the analysis of the area of snow cover in the LANDSAT snow cover scene. And, finally, using knowledge guided segmentation, determine the change in the number of certain objects in the urban or industrial scene.

### Segmentation

The first step towards the generation of a symbolic description of an image is to divide continuous image signal into discrete components sharing similar properties. Our work on segmentation is an extension of the histogram guided region splitting technique developed by Ohlander (1975). This method was originally developed for use on color images. Basically the procedure splits a region into subregions thresholding one of the spectral inputs. The threshold is selected by the analysis of the histograms of

the values for all pixels in the region (one histogram for each spectral input). The threshold values are selected as the upper and lower bounds of the "best separated" peak which appears in the set of histograms. There are two problems in the use of this technique for the segmentation of our set of images. First, the segmentation method is much too slow for processing a large set of images in a reasonably short time. Second, the segmentation technique was developed for multi-spectral images and could not be expected to work as well on the monochromatic images.

Planning: The first problem is solved by the introduction of "planning." By planning, we mean the generation of an approximation for the final segmentation using a reduced version of the image and the use of this approximation as a plan to more efficiently derive the true segmentation of the image. Ohlander gave a time of about ten hours for the segmentation of a color image with 0.5 million pixels (nine parameter for each pixel, each parameter represented by about eight bits). This time would be reduced to about five hours, if run now, because of simple algorithmic improvements to many of the programs which he used (such as to the smoothing procedure). The use of planning further reduces the total time to less than one half an hour (including the reduction time), or about one order of magnitude. There is also overhead involved in the manipulation of large images which is not reflected in these times. We present the segmentation times in hours rather than the number of operations which was used elsewhere in this thesis to enable the comparison with the times-for Ohlander's segmentation. Both of these segmentation systems were run on the same computer system, so that the times are comparable.

Monochromatic Images: The segmentation of monochromatic images required additional alterations to the initial segmentation method. The original segmentation method was based on the hope that if one feature can not provide a reasonable split of the region, then, perhaps, some other color feature. For example if two regions have the same intensity but are different colors, then the intensity parameter alone could not be used for segmentation, but another color parameter (possibly hue or Q) will help in the segmentation. When the procedure is presented with only one spectral input, there is no other color parameter to turn to when there is only one peak in the histogram. The large monochromatic images also contained many small different objects which caused the histogram to have only one peak since the range of intensities for each region overlapped the ranges of intensity for other object.

We can introduce additional spectral-like features by the use of simple textural operators designed to show specific features such as homogeneous regions, or high contrast areas. We introduced a feature, the number of micro-edges in the reduction window, to indicate general homogeneous regions. A homogeneous region is one with few micro-edges so that these regions can be extracted by using a threshold of zero edges in the plan image. The points where the few edges occur will appear as small holes in the segmented region and are eliminated by the smoothing process. This threshold could not be applied directly to the initial micro-edge image (it is a binary image). The individual micro-edges would appear as small holes (a few points) in the thresholded image and would be swallowed up in the homogeneous region by the refining and extraction procedures. The smooth regions generated



by the plan limit the area where this threshold is applied so that only a small number of edge points are swallowed up. The regions which are extracted are more sensitive to noise in the image, especially noise in one part of the image such as scratches. This feature is not generally useful for the extraction of exact regions, but proved useful for the extraction of general homogeneous regions.

Another textural measure is the excursion of intensity values in the reduction window (maximum in the window minus minimum in the window). This measure is applied to the SLR scene to distinguish between the high contrast areas and the low contrast areas. This textural measure generated large general regions which correspond to the general textured areas. These were the only textural operators which were used in the segmentation of images.

Many other operators are possible, and for easy incorporation in a general segmentation method the operators should produce image like values for all points in the image. There are many possible textural operators, but we did not want to turn this thesis into an exploration of all possible texture operators. The intent was not to judge the quality of other textural operators, but it should be easy to incorporate others into this system.

Another technique used in the computation of histograms was to use portions of the image rather than for the entire image. This was intended to approach a solution to the problem of many small similar regions. The use of partitions means that the number of separate objects which contribute to one histogram is reduced. If the partitioning of the image is continued as far as possible, at some point there will be only two distinct regions (or possibly one) to contribute to the histogram. At this point the threshold for segmentation would be obvious. Going to these extremes should not be necessary. We implemented a division of the image into only four or nine partitions.

#### Feature Extraction

There are at least two very different techniques to give a symbolic representation of an image. One is a three-dimensional description of the objects in the scene such as representing all objects by a set of simple three-dimensional objects. This representation did not appear to be feasible to derive from general multiple views, and did not appear to be very useable for change analysis. We decided that the symbolic description would be composed of a set of regions which would be those generated by the segmentation procedure, and a set of features for each region describing various properties of the region. We group the features into classes similar to those used by human beings performing the same sort of tasks. These feature classes include size, shape, color (including texture), location, and patterns. The exact feature measures were designed to capture various aspects of these feature classes. We computed the region size, absolute position of the center of mass, the position relative to other regions (above, below, etc.), adjacencies, average of color values or textural values, orientation, orientation independent length to width ratio, the fraction of the minimum bounding rectangle filled by the region, and the  $\text{perimeter}^2/\text{area}$  designed to indicate irregular regions. These are not all the feature measures which might be necessary for other tasks, and results should be more reliable when more features are available.

The methods for the computation of these features were not optimized. The computation effort for some

features is insignificant since these features are derived from other values (such as the length to width ratio, the orientation, fractional fill,  $\text{perimeter}^2/\text{area}$ ). The expensive operations were the ones performed on all the points in the region, where most of the expense was in looking at the region points rather than computing feature values. The expensive features included: the color averages (mostly because they are used so often rather than being individually expensive), boundary computations (though it is less expensive than color averages since fewer points are accessed), orientation transformation computations (since they use the boundary computation), the initial color and texture transformations, and the relative position computations (which are expensive only because of the machine implementation). Like the segmentation operations, the expensive feature computations are amenable to implementation on special processors. The major descriptive feature which we did not study is the extensive use of textural measures.

#### Symbolic Registration and Change Analysis

The earlier systems for change analysis relied on correlation guided matching to locate corresponding point pairs and used the location differences of these point pairs either for transforming one image so that it is aligned with the other, or for depth analysis of stereo images. The aligned images are subtracted, producing a third difference image. This difference image must then be analyzed to determine where the changes occurred, and what type of changes occurred. Special purpose systems have been built to perform these tasks, so that these apparently expensive operations are performed quickly. Change analysis systems which are intended to operate on uncontrolled image pairs (i.e. not stereo pairs) encounter several problems. The addition of more color parameters makes the problem more complex since the extra spectral inputs must be processed just like the initial input instead of simplifying the processing. Major changes in the point of view of the observer (especially in oblique views) will cause objects to change position with respect to each other and can cause inaccurate matches when those matches depend on intensity values in a neighborhood and are difficult (if possible) to account for in a global warping of the image. These systems used a "rubber sheet" warping so that points adjacent in one image are assumed to be adjacent in the other image. A new object in the scene can cause errors in matching, but such changes would usually be indicated as large differences in the difference image.

We present symbolic matching as an alternative matching technique to eliminate the problems encountered by earlier signal based change analysis methods. The addition of extra spectral inputs makes the segmentation processing easier and more reliable, and, if the desired regions are large enough, the use of planning means that the segmentation times will not be adversely affected by the addition of more inputs. Also, the addition of color parameters means that the matching procedures will have more features to use in the matching, this should also improve the reliability of the symbolic registration. Since the matching for one region does not necessarily depend on the intensity values in the image adjacent to the region being matched, the change in the relative position of objects should not reduce the chances of a correct match. We use several different features of the region including the adjacency and relative position relations, but the knowledge

about the scene can specify that the relative position or adjacency relations will change: thus indicating that these features are not used for the symbolic registration. New objects are indicated by regions in the second image which had no corresponding region in the first image, and missing objects by regions which fail to match with any region. Finally, the change results produced by a signal based change analysis system are in the form of another image and must then be processed again to determine what changes have occurred. The symbolic change analysis system describes the changes as changes in the features of regions (or changes in the number of occurrences of an object). Thus there is no need for extensive processing of the resulting image to discover the kinds of changes, since these changes are given directly from the symbolic analysis.

**Symbolic Registration:** We developed a procedure which will determine a match rating for two regions in two different images. This rating procedure incorporates the differences between all available features of the regions. If the match is exact (e.g. matching a region with itself) then the rating will be zero, and as the match worsens, the rating decreases. The knowledge sources can indicate that certain features will change and thus should not be used in the matching procedure. For example, when the task description indicates that there are rotation differences between the two images, the matching procedure will not use the rotation dependent features such as the absolute position, the orientation, regions above, regions below, etc. Rather than eliminate the use of these features altogether, we introduce different strengths for features which should remain constant and features which will change. The strengths are selected so that a bad match in one feature that should remain constant will have more impact than several bad matches in features which may change. This region to region match procedure is used in the symbolic registration procedure to find the best available match. To find the region in the second image which corresponds to a region in the first image, the symbolic registration procedure matches each possible pair of regions to find the best match. This best match is considered to be the corresponding region. Even if a region does not have a corresponding region in the other image, some region will be selected as the corresponding region. This region will be the most similar region, but these two regions should have differences in features which should remain constant. Also, another region in the first image should correspond to the same region in the second image. This matching procedure has been applied to the six sets of images. We generated about a dozen sets of symbolic matching results (because we can match the second image to the first image in addition to matching the first image to the second image we can generate more sets of matching results than we have scenes). Several images including monochromatic and side-looking radar images were registered using this technique. Figures 1 and 2 illustrate the nature of the results obtained by this technique in the presence of changes in perspective and scale.

**Change Analysis:** For some images we are given (through the image description) the fact that there is a scale change between images (as in the urban-industrial scene). The amount of the scale change is not given by the knowledge elements, but it can be computed from the size differences found in early matches. This scale change is used to adjust the size measures for regions in later matches. Since there is a scale difference between the two

images, the absolute size and location features will change and can not be used as constant features in the matching operation. But with the use of the computed scale difference, the size feature can be used as if it is a constant feature. This use of the change results derived from the initial symbolic matching procedure can also be applied to the absolute location and orientation features, in addition to the size feature. These adjustments can apply only when the changes are uniform throughout the image, which is not the case when there are perspective changes as in oblique views. But such adjustments are possible to use in most aerial images.

The pier subsection was used for the analysis of the changes to determine the number of "ship" regions in the two images. To perform this task we generated a pseudo-image containing a representative ship, pier, water, and shadow region. We then matched the two pier area segmentations with this pseudo-image to determine which regions are "ships." The results of this analysis are shown in Figure 2. Some of the "ships" were incorrectly segmented: they were broken into two regions, or only half of the region was segmented and the other half was merged with other regions (such as the piers or water). But the ship regions which were segmented were matched to a ship region. In the first subsection some errors occurred. The water regions were not as smooth and thus the average intensity and number of micro-edges in some of the water areas resembled the ship parameters more than the water parameters. In the second subimage errors were also caused by the matching of small parts of piers to ships because of the number of micro-edges in these pier regions. This was an attempt to extend the matching procedure into a rudimentary "recognition" procedure to compute the number of occurrences of a type of region feature.

The symbolic registration and change analysis processing is relatively fast when compared with all the other processing. This processing is best suited for implementation on general purpose computers rather than special purpose processors.

### Conclusion

In this paper we have described the structure and present state of performance of a symbolic registration and change detection system. This appears to work well in a wide variety of images in finding corresponding regions. The match is based purely on symbolic features and the system clearly demonstrates the feasibility of image registration including cases where signal registration techniques would have failed either because of wide differences in the direction and position of view (See also the paper by CDC in this volume). The concept of change analysis, on the other hand, proved to be more elusive. It appears to be extremely task dependant i.e. one has to impose substantial task structure and correspondingly highly task specific programs before one can have useful change detection systems. At present, work is in progress to extend these techniques to a larger class of images with more severe perspective and scale changes.

### References

- R. Ohlander (1975). "Analysis of Natural Images," Ph.D Thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA. (April 1975).
- K. Price (1976). "Change Detection and Analysis of Multi-Spectral Images," Ph.D Thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA. (Dec. 1976).



AN EXAMPLE OF SYMBOLIC REGISTRATION IN THE  
PRESENCE OF PERSPECTIVE CHANGE

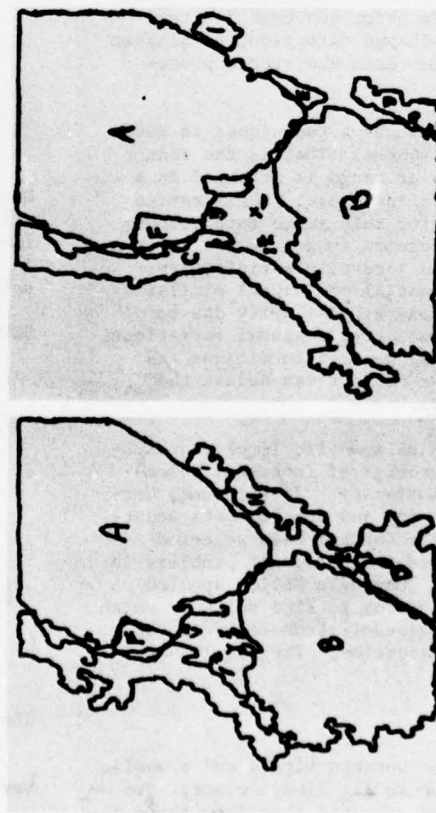
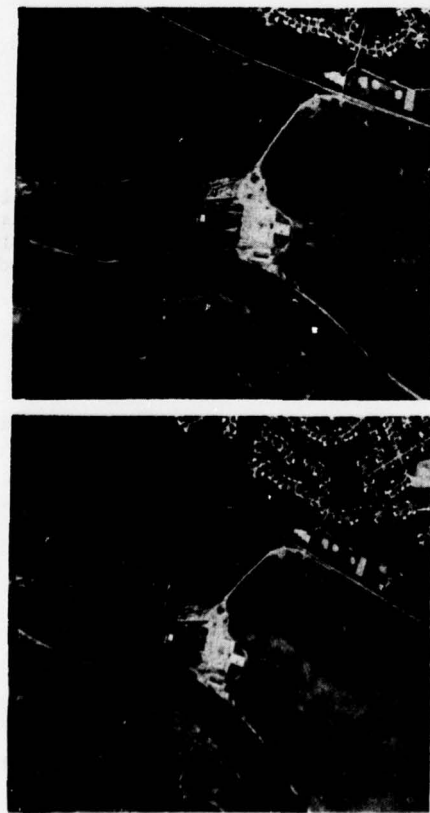


FIGURE 1: The Power Plant scene illustrates symbolic matching of two images taken from different perspectives. Due to the viewing aspect, the size and shape of corresponding regions in these images differ. These images can be characterized as having large smooth areas and few bright regions. Regions A and B are the regions above and below the power plant. Region C is the river to the left of the power plant. These regions are correctly matched due to their size and position with respect to each other in spite of the changes in shape due to the change in perspective.

AN EXAMPLE OF OBJECT LOCATION AFTER  
SYMBOLIC REGISTRATION

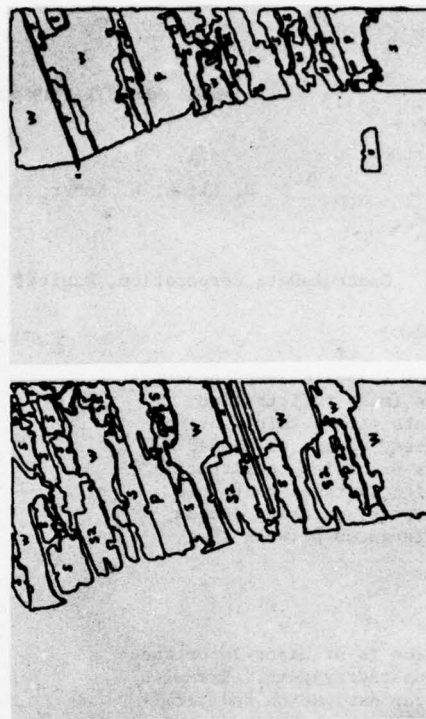
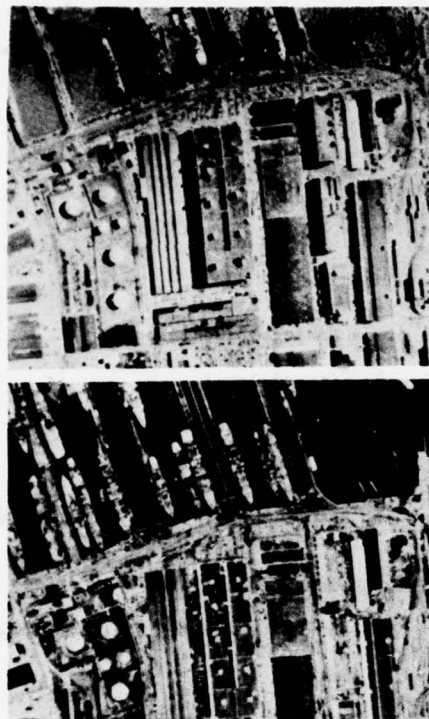


FIGURE 2: The Norfolk scene task involves the detection of new or missing objects in a given area of the scene where the images exhibit a scale change. This requires limiting the area of the two images being analyzed, and determining size and position differences between objects in the images. The task is to determine the number of ships in the dock area on the right hand side of the image. We determine whether a region is water, a ship, or a pier, rather than whether the region matches another unidentified region.



## IMAGE REGISTRATION EXPERIMENTS

B. Glish, W. Kober, G. Swanlund

Control Data Corporation, Digital Image Systems Division

## ABSTRACT

The usefulness of symbolic level processing was evaluated for image registration. Coarse alignment points were obtained using segmentation procedures. These were used as coarse alignment of a precision registration technique. The results indicate that this procedure is effective for matching images, which have gross differences between them.

## INTRODUCTION

Image registration is of major importance in many imagery processing systems. Examples are; scene matchers for navigation and terminal guidance, overlay of reconnaissance and reference data bases for target position transfer, registration to prior coverage for temporal analysis and change detection, overlay or fusion of multisensor data and stereo photogrammetry.

The principal matching techniques to date are signal level matchers. That is the sensor signal of intensity or range is arranged in a two dimensional array or image. Registration is achieved by driving this image into correspondence with a reference image. While these techniques have been generally satisfactory, they require both spatial and signal similarity between images. Gross dissimilarity due to different viewing geometry, seasonal variations, sun angle variations, weather conditions and changes in the scene content can defeat the registration process.

Recent efforts on symbolic level representations offer the promise of increased tolerance to image dissimilarity. To this end, Carnegie Mellon University and Control Data undertook a joint effort. Control Data selected imagery which represented difficult problems in image registration. Carnegie Mellon applied symbolic level procedures to find suitable match parameters and subsequently transferred these to Control Data for evaluation. The results are described below.

## DEFINITIONS

The distinction between signal and symbolic level representation is not always clear. For the present study we arranged them into three

groups of ascending abstraction.

1. Raw signal level
  - Intensity (or tonal) image
  - Range image
  - Intensity vs. range (three dimensional) image
2. Enhanced signal level
  - Spatial frequency filtered image
  - Edges
  - Intensity or tonal slice
3. Symbolic level
  - Tonal regions
  - Texture regions
  - Object shapes

Group 2 might be considered symbolic level but principally is only a deletion of portions of the image. Furthermore, these features are already in wide spread use and hence are outside the spirit of this effort.

## STUDY APPROACH

The application is restated as:

Precision matching of two images of a scene in spite of:

- Seasonal change
- Illumination change
- Ground change
- Viewing geometry change
- Different sensor types

The problem is:

Signal level matchers tolerate only limited variation between images.

The approach is:

Utilize symbolic level representations for coarse match and signal level for precision matching.

The approach is illustrated in Figure 1.

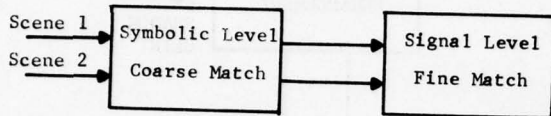


Figure 1. Approach

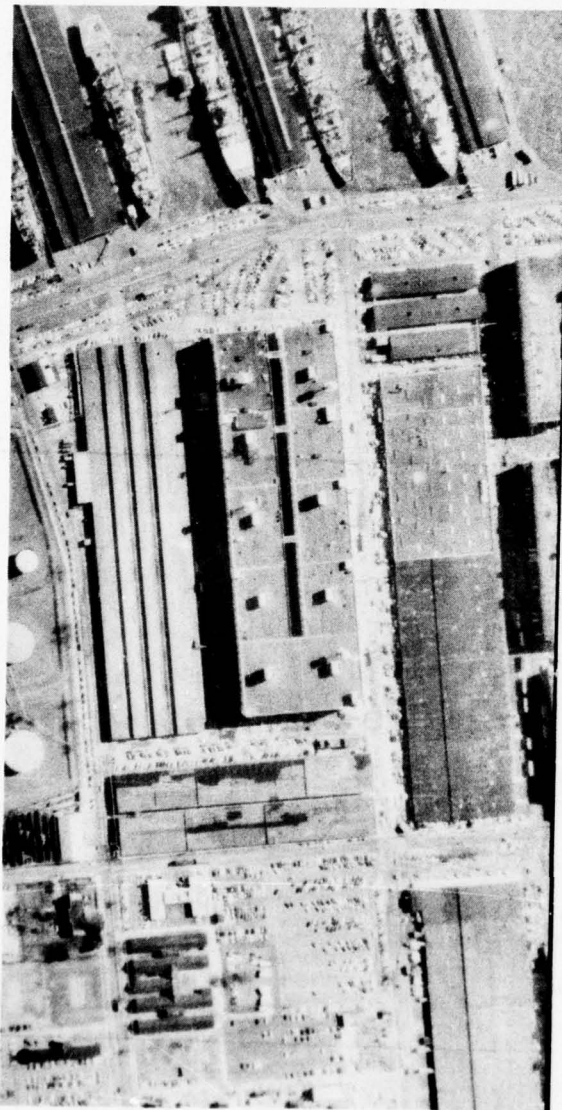


Fig. 2 Image 1



Fig. 3 Image 2

#### APPLICATION: CHANGE DETECTION

One application requiring precision registration is change detection. An example of a difficult case is the Norfolk Harbor scene. The two images of this scene are shown in Figure 2 and 3. They were taken under the following conditions:

FACTOR	IMAGE 1	IMAGE 2
DATE	28 Jan 71	14 Apr 72
TIME	1310	1520
ALTITUDE	4800'	5700'
SCALE	1:9600	1:11,400
SHADOWS	Yes	No
GLINT	No	Yes

GROUND CHANGES: VEHICLE MOVEMENT, CONSTRUCTION

Segmentation processing using tonal and texture features by S. Price<sup>1</sup> at CMU provided the symbolic level feature maps shown in Figure 4. His processing included a search which identified 12 matching features between the two images. The centroids of these 12 were then used as the coarse registration parameters.

The fine registration procedure used was a precision correlation algorithm called TRAK developed previously at CDC<sup>2</sup>. TRAK allows control points and scaling parameters to be inserted for coarse alignment. Thus it was relatively easy to incorporate the output from the symbolic level coarse matching. Two alignment parameters were computed, 1) the most extreme points in x and y coordinates were used to estimate scaling and rotation between the two images, 2) two other points were used to estimate translation errors. With these inputs, precision registration was evaluated for various options within the TRAK routine. These options are listed in Figure 5. Shadow and glint compensation remove these decorrelating influences from the correlation.

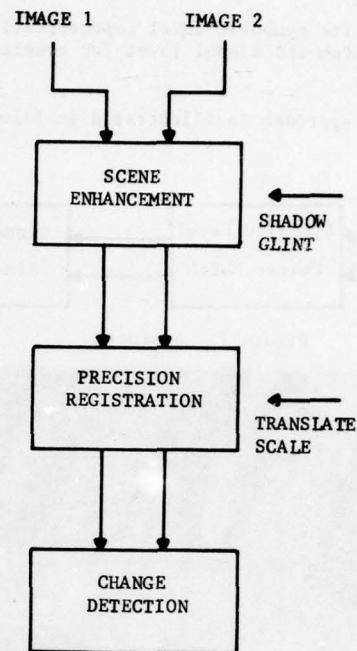


Fig. 5

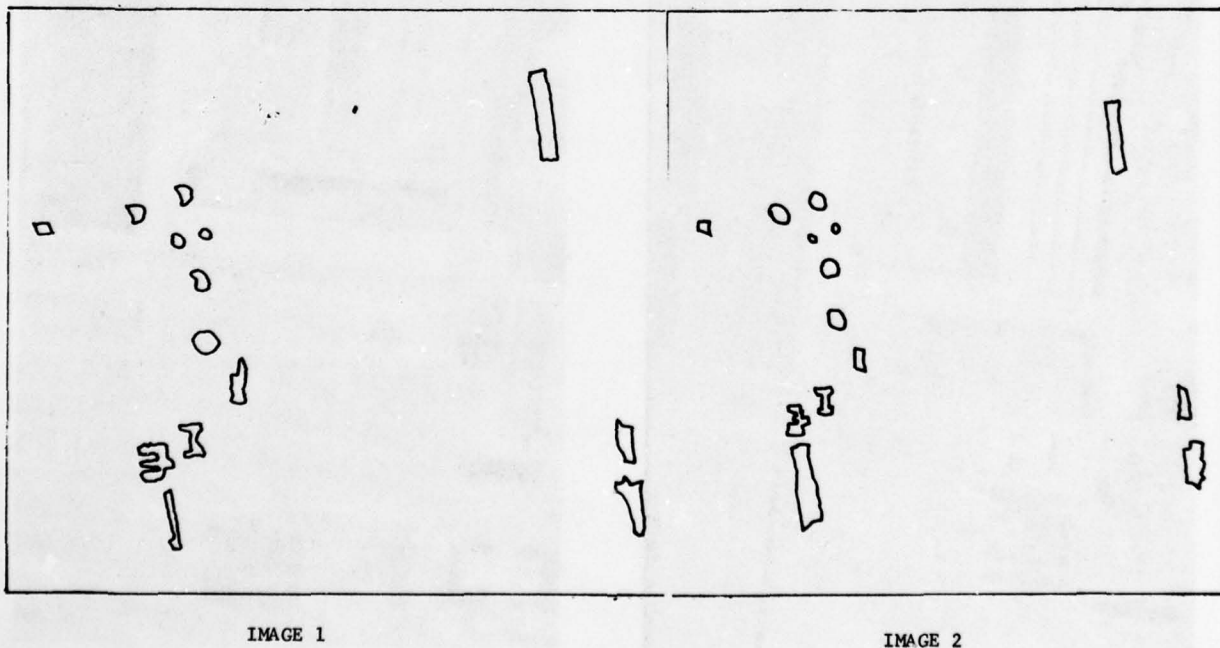


Fig. 4. Match Regions From Symbolic Level Process



## EXPERIMENTS

Three experiments were conducted using the same coarse registration points for acquisition and scaling.

- 1) Precision registration of the original two images.
- 2) Precision registration with shadows and glint deleted in the registration process.
- 3) Precision registration after pre-processing to remove redundancy.

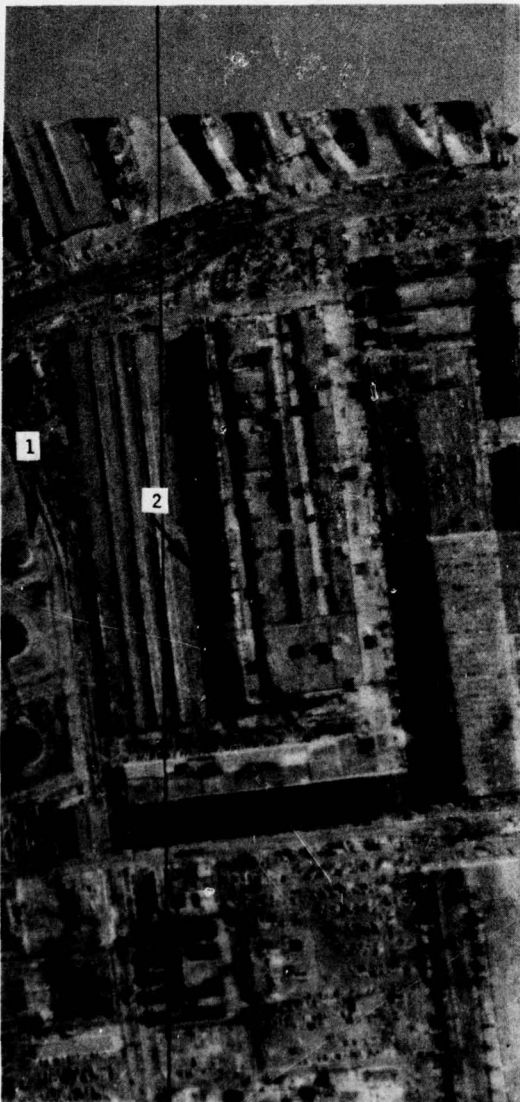


Fig. 6. Difference Image

The difference image after the precision registration of the two images, Fig. 2, 3 is shown in Fig. 6. The initial registration points were at the left side of the image, in the vicinity indicated by arrow 1.

The initial registration points had some translation error as evidenced by a slight ghosting (light ring) on the right edge of the oil tanks. The precision registration algorithm recovered and soon had the images in registration. However, it then encountered regions with extensive shadowed portions in image 1 which were not present in image 2. This caused the process to diverge. By the time it reached the point indicated by arrow 2 it had diverged beyond recovery. At least two options are available at this juncture. 1) Include more points from the segmentation process so that the precision registration process can be re-initialized or 2) detect the shadow and glint regions, and delete them from the registration process. The first option of restart was not practical in this case since there weren't any control points available in the region of poor similarity.

The second option of shadow and glint removal was used. The results of this process is shown by the difference image, Fig. 7. Note that the shadow and glint regions are retained in the difference image. They were only left out during the registration process. With this option, the images were kept in registration over the entire scene. This can be verified visually by examination of the difference image. For example, the arrow at point 3 shows a distinctive region on the roof of a building. This was dark on both images. Thus the difference image should be close to the grey level corresponding to zero difference. This is the case.

It should be noted that the registration procedure also normalizes the tonal distributions between images. Thus regions of different tonal values would be equalized before differencing. The region used for adjustment must be selectable so as not to normalize out desired changes. Thus it is both an image - and target - dependent parameter.

Fig. 8 is the same difference image as Fig. 7 but thresholded into positive (black) negative (white) and no (grey) change regions. The selection of threshold determines the interval assigned to no change. These are changes which have some level of contrast to the background. Visual inspection reveals changes due to shadows (arrow 4), vehicle movement (arrow 5), ship movement (arrow 6) and construction (arrow 7). It is significant that the precision registration and tonal normalization permits a confident analysis of changes due to the movement of small objects and to changes of low contrast. Furthermore, the shadows can be deleted if they are distracting to the interpretation.

## PREPROCESSING

One form of preprocessing often used is

spatial filtering. For example, low pass filtering can permit more tolerance to out-of-registration errors for an initial alignment procedure. The image is re-mapped and then a more precise registration is attempted. On each successive run, the filter bandwidth is increased.

The Norfolk image was run with simple low pass filtering. However, the registration results did not improve. The apparent reason for this behavior is that much of the similarity between images was in the edges. Filtering these edges destroyed a significant amount of the correlation signal content.



Fig. 7. Difference Image Using Shadow Deletion

Another preprocessing technique tried was an adaptive non-linear filter. Here the procedure retains edge discontinuities while smoothing regions with no little tonal variation. The result of registration with this preprocessing is shown in Fig. 9 as a thresholded difference image. For comparison, Fig. 10 shows a thresholded difference image without this preprocessing. The results are dependent on the threshold setting but in general appear similar. The registration precision was about the same. The major difference seems to be that the preprocessing removes some of the noise while also losing some of the change detail. Scene dependent knowledge is needed to determine if the detail discarded is significant.

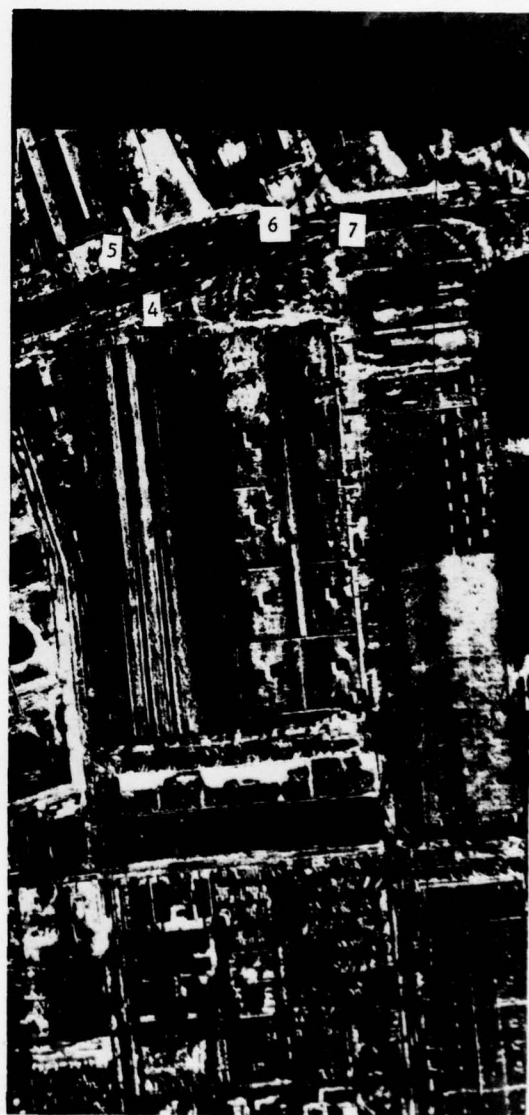


Fig. 8. Thresholded Difference Image



Fig. 9. Thresholded Difference Image With Preprocessing

One conclusion from this last experiment is that the adaptive procedure gave better registration performance than simple low pass filtering. However, it needed shadow deletion to achieve registration. Thus the performance was not any better than without preprocessing.

#### SUMMARY OF EXPERIMENTS

The symbolic level processing provided scale and translation information for coarse alignment. Subsequent precision registration was successful only if the shadows were deleted in the registration process. In this case, precise registration was obtained in spite of gross changes in scene content and lighting conditions.



Fig. 10. Difference Image Without Preprocessing

Preprocessing by low pass filtering gave poorer registration performance. A preprocessing technique which retained detail gave similar performance to the unpre-processed case. This illustrates a well known principle that preprocessing is sensitive to scene content as well as to the amount of misregistration.

#### REFERENCES

1. Price, K. E., "Change Detection and Analysis in Multi-spectral Images", Dept. of Computer Sciences, CMU, 18 Dec. 1976.
2. Lillestrand, R. L. "Techniques of Change Detection", IEE Transactions on Computers, Vol. C-21, No. 7, July 1972.



## IMAGE SEGMENTATION AND OBJECT DETECTION BY A SYNTACTIC METHOD

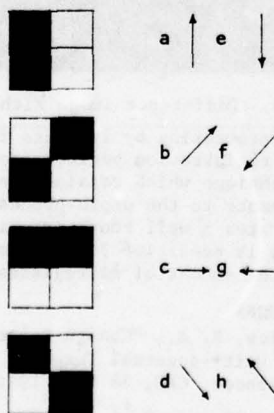
Janmin Keng  
Purdue University  
W. Lafayette, Indiana 47907

## INTRODUCTION

Most of the existing image segmentation techniques utilize only statistical properties of an image and ignore the useful syntactic and contextual information. Since an image often exhibits a hierarchical structure, image segmentation can be effected by the syntactic approach. This paper describes a syntactic image segmentation technique which detects the edges of small textural areas as well as larger ones from the real world satellite and aerophotographic images. This technique has been extended to the object detection. The experiments on tactical target detection from the infrared images have been conducted. The experimental results show that this technique is general and feasible to different types of images for image segmentation and object detection.

## GRAMMATICAL INFERENCE

In Keng and Fu [4], and Fu and Booth [2] the details of grammatical inferences for tree grammar and tree transformational grammar are described. Here we state the inferred results as follows. The inferred tree grammar is to describe the boundaries of the image segments. The primitives for those patterns are:



and the window size is chosen as 8x8 array of pixels. The positive samples are those patterns starting from a primitive followed by at most three branches. The negative samples are those

patterns in which there is no boundary line or just singular primitives or pixels in it. Applying the tree grammatical inference procedure, a set of tree grammar is inferred to describe the boundary structures.

The tree grammar is  $G_t$

$$G_t = (V, r, P, S)$$

where  $V = \{S, \frac{1}{2}, A_1, A_2, A_3, A_4, A_6, A_7, a, b, c, d, e, f, g, h\}$

$$r(a) = \{0\} \quad r(b) = \{1, 0\} \quad r(c) = \{3, 2, 1, 0\}$$

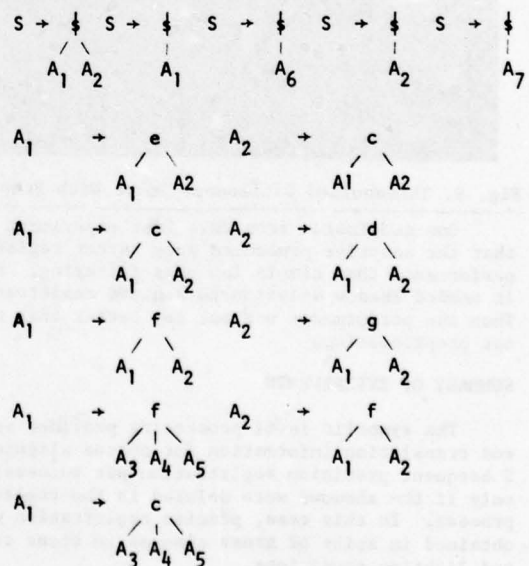
$$r(e) = \{2, 1, 0\} \quad r(f) = \{3, 2, 1, 0\}$$

$$r(g) = \{2, 1, 0\} \quad r(h) = \{1, 0\}$$

$$r(\frac{1}{2}) = \{2, 1\}$$

$$V_T = \{+a, \times b, +c, \times d, +e, \times f, +g, \times h, \frac{1}{2}\}$$

and grammar rules  $P =$



$A_3 \rightarrow b \quad A_3 \rightarrow d \quad A_3 \rightarrow e$   
 $\quad \quad \quad | \quad \quad \quad | \quad \quad \quad |$   
 $\quad \quad \quad A_3 \quad \quad \quad A_3 \quad \quad \quad A_3$   
 $A_3 \rightarrow f \quad A_3 \rightarrow g \quad A_3 \rightarrow c \quad A_3 \rightarrow h$   
 $\quad \quad \quad | \quad \quad \quad | \quad \quad \quad | \quad \quad \quad |$   
 $\quad \quad \quad A_3 \quad \quad \quad A_3 \quad \quad \quad A_3 \quad \quad \quad A_3$   
 $A_4 \rightarrow b \quad A_4 \rightarrow d \quad A_4 \rightarrow e$   
 $\quad \quad \quad | \quad \quad \quad | \quad \quad \quad |$   
 $\quad \quad \quad A_4 \quad \quad \quad A_4 \quad \quad \quad A_4$   
 $A_4 \rightarrow f \quad A_4 \rightarrow g \quad A_4 \rightarrow c \quad A_4 \rightarrow h$   
 $\quad \quad \quad | \quad \quad \quad | \quad \quad \quad | \quad \quad \quad |$   
 $\quad \quad \quad A_4 \quad \quad \quad A_4 \quad \quad \quad A_4 \quad \quad \quad A_4$   
 $A_5 \rightarrow b \quad A_5 \rightarrow d \quad A_5 \rightarrow e$   
 $\quad \quad \quad | \quad \quad \quad | \quad \quad \quad |$   
 $\quad \quad \quad A_5 \quad \quad \quad A_5 \quad \quad \quad A_5$   
 $A_5 \rightarrow f \quad A_5 \rightarrow g \quad A_5 \rightarrow c \quad A_5 \rightarrow h$   
 $\quad \quad \quad | \quad \quad \quad | \quad \quad \quad | \quad \quad \quad |$   
 $\quad \quad \quad A_5 \quad \quad \quad A_5 \quad \quad \quad A_5 \quad \quad \quad A_5$   
 $A_6 \rightarrow e \quad A_6 \rightarrow e \quad A_7 \rightarrow c \quad A_7 \rightarrow c$   
 $\quad \quad \quad | \quad \quad \quad | \quad \quad \quad | \quad \quad \quad |$   
 $\quad \quad \quad A_6 \quad \quad \quad A_1 \quad \quad \quad A_7 \quad \quad \quad A_1$   
 $A_1 \rightarrow e \quad A_1 \rightarrow c \quad A_1 \rightarrow f$   
 $\quad \quad \quad | \quad \quad \quad | \quad \quad \quad |$   
 $\quad \quad \quad A_1 \quad \quad \quad A_1 \quad \quad \quad A_1$   
 $A_2 \rightarrow c \quad A_2 \rightarrow d \quad A_2 \rightarrow g \quad A_2 \rightarrow f$   
 $\quad \quad \quad | \quad \quad \quad | \quad \quad \quad | \quad \quad \quad |$   
 $\quad \quad \quad A_2 \quad \quad \quad A_2 \quad \quad \quad A_2 \quad \quad \quad A_2$   
 $A_1 \rightarrow e \quad A_1 \rightarrow c \quad A_1 \rightarrow f$   
 $A_2 \rightarrow c \quad A_2 \rightarrow d \quad A_2 \rightarrow g \quad A_2 \rightarrow f$   
 $A_3 \rightarrow b \quad A_3 \rightarrow d \quad A_3 \rightarrow e$   
 $A_3 \rightarrow f \quad A_3 \rightarrow g \quad A_3 \rightarrow c \quad A_3 \rightarrow h$   
 $A_4 \rightarrow b \quad A_4 \rightarrow d \quad A_4 \rightarrow e$   
 $A_4 \rightarrow f \quad A_4 \rightarrow g \quad A_4 \rightarrow c \quad A_4 \rightarrow h$   
 $A_5 \rightarrow b \quad A_5 \rightarrow d \quad A_5 \rightarrow e$   
 $A_5 \rightarrow f \quad A_5 \rightarrow g \quad A_5 \rightarrow c \quad A_5 \rightarrow h$   
 $A_6 \rightarrow e \quad A_7 \rightarrow c$

A transformational grammar is a set of grammar rules to transform one pattern from one form to another [1,3,4]. A line smoothing technique can be designed by a transformational grammar. Here we introduce the tree transformational grammar for line smoothing technique. The concept of the syntactic line smoothing technique is as follows: The irregularities are usually caused by the digitizer, noisy patterns, and so forth. These are in forms such as zig-zag of the line patterns. The tree transformational grammar evaluates the contextual information of the patterns.

If the context of the pattern satisfies the transformational grammar, that patterns is transformed into a smoother pattern. By this syntactic line smoothing technique, the zig-zag of lines is smoothened.

#### SYNTACTIC IMAGE SEGMENTATION ALGORITHM

A syntactic approach to image segmentation has been investigated which involves two levels of processing. The first level, referred to as the preprocess, primitive extraction, consists of two steps referred to as (1) texture region primitive extraction, and (2) boundary primitive extraction. The second level, which is the syntactic analysis, requires inference of tree grammar to describe the boundaries of homogeneous regions. The grammatical inference procedure has been described in section 11. The process of tree grammar analysis utilizes the corresponding parser from the inferred tree grammars to process the primitive extracted image. Then an image is segmented.

Algorithm SISA (Syntactic Image Segmentation Algorithm)

Preprocess: Primitive Extraction

A. Texture region primitive extraction

Since the satellite image is very rich in texture, the simple grey level thresholding technique for segmentation is inadequate. Texture analysis has been studied in [5,6]. The first part of the proposed technique is texture region primitive extraction. The texture is defined as the over-all or average spatial relationship which the grey levels in images have to one another. The histogram equalization technique is applied first to requantize the image. If there are finite grey levels  $k$ , the joint probability density of the pairs of grey levels that occur at pairs of points with distance  $d$  is computed. The array is a  $k$  by  $k$  matrix  $P(i,j)$ . Then a variability texture feature is calculated to measure the spatial relationship of the grey levels of an image.

$$\text{VARIABILITY} = - \sum_{i,j} \sum_R \left( \frac{P(i,j)}{R} \right) \log \left( \frac{P(i,j)}{R} \right)^k. \quad R \text{ is}$$

the normalization constant and  $k$  is the range constant. From our experiment on the images of the Indianapolis area (central Indiana), this texture measurement characterizes the major land-use classes as agriculture, tree, old residential, new residential, and water areas.

The window size used is  $11 \times 11$  pixels. Note that it is quite possible for some textural area to be smaller than the window size, or the boundary between different textural areas lies in the operation window. In the technique, this is taken care of as we locate those boundaries by moving the  $11 \times 11$  operation window 4 pixels at a time. Thus, the potential boundaries could be preserved and the spatial relationship is still

extracted because the window size (11x11) was not reduced. After obtaining the texture values for these 4x4 unit cells, we threshold the histogram of texture values in the texture domain, and then assign texture codes to the segments. (The histogram is made by those values from shifting the 11x11 window 11 pixels at a time). In the experiments on the test images, since the histogram is strongly multimodal, it is reasonable to assign different codes to the pixels to form the texture region primitives.

### B. Boundary primitive extraction

1. Horizontal processing: Following the texture region primitive extraction is the boundary primitive extraction. The Horizontal Process processes the "texture region primitive assigned image" row-wise to locate the potential horizontal boundary segments. The operation procedure is as follows: let  $Q(I,J)$  be the picture function at location  $(I,J)$

Step 1. Start with  $Q(I,J)$  as reference.

Step 2. Compare  $Q(I,J)$  with  $Q(I,J+1)$ . If the distance is smaller than a specified value, a "zero" is set on  $Q(I,J)$  and  $Q(I,J+1)$ . Then  $Q(I,J)$  and  $Q(I,J+2)$  are compared. If the distance is greater than or equal to the specified value, A "one" is set for  $Q(I,J+2)$  as a potential boundary primitive. Then the same process is applied with  $Q(I,J+3)$  as the reference.

Step 3. When this process is operated to the rightmost of the row the  $Q(I+1,J)$  is the reference and Step 2 is applied until all the rows are processed.

The idea of this process is to treat the image matrix as independent rows. After this process the potential vertical boundaries of the image are detected. The reason for comparing  $Q(I,J)$  and  $Q(I,J+2)$  (when  $Q(I,J+1)=0$ ) in Step 2, is because the reference must be kept in the same operation. If instead of comparing  $Q(I,J)$  and  $Q(I,J+2)$ , the  $Q(I,J+1)$  and  $Q(I,J+2)$  are compared. The reference is shifted, thus none of the boundary primitives will be detected.

2. Vertical processing: The Vertical Process is in the same manner as the Horizontal one except that it processes the image column-wise to locate the potential horizontal boundary segments.

3. Logic integration: The result of horizontal process is defined as  $H$  and the result of vertical process is defined as  $V$  in the Boolean algebra. The Logic Integration is a Boolean function of  $H$  and  $V$  and it is defined as  $R(H,V)$ .  $R(H,V) = H+V$ .

4. Syntactic line smoothing: A tree transformational grammar is designed to reduce irregularities and smooth the patterns. The grammatical inference scheme has been described in detail in section 11. [4].

### SYNTACTIC ANALYSIS

In the areas of language compiling, computer communication and syntactic pattern recognition, error correcting parsing techniques have been applied to remove the uncertainty and errors in the process. The syntactic analysis for image segmentation is a top-down tree parser with the error correcting ability. The parser differs from other error correcting parsers in the way of grammar construction and the parsing scheme. The overall system has an on-line syntactic tree parser and an off-line error correction mechanism. The advantage of this design is the high efficiency when the input pattern is not noisy. Because the off-line error correction is initiated and used only when the uncertainty occurs. The on-line tree parser is a top-down parsing scheme which accepts the correct patterns and reject the incorrect patterns. The uncertain patterns are left for the error correction mechanism to correct. Thus, the grammar used is a grammar set which does not describe the error transformation. The error correction mechanism is a top-down error correction scheme, also is the grammar parser. The top-down error correction is designed because in parsing tree languages, the partially constructed tree conveys much usable information about what should appear next in the structure. This information is not as readily available in the bottom-up parsing. The bottom-up method cannot easily use the global context (like all incomplete branches), and will have to rely on the local context immediately surrounding the error pixels. The inputs for this top-down tree parser are the tree languages, which have been encoded as tree linked lists, (shown in the section of syntactic line smoothing technique) [4] and the tree grammars. The tree grammars are in the data structures of grammar table which is shown as follows:

For a tree grammar  $G_t = (V, r, P, S)$ ,

$$V = \{S, V_1, V_2, \dots, V_n, t_1, t_2, \dots, t_m, \frac{1}{2}\}$$

$$V_T = \{\frac{1}{2}, t_1, t_2, \dots, t_m\}$$

$$r(t_1) = k_1, \dots, r(t_m) = k_m, r(\frac{1}{2}) = k_0$$

$$P = S \xrightarrow{\frac{1}{2}} \text{ and so forth.}$$

$$\begin{matrix} \frac{1}{2} \\ \swarrow \quad \searrow \\ V_1 \quad \dots \quad V_k \end{matrix}$$

the maximum value of  $\{k_1, \dots, k_m\}$  is assumed to be  $k$ , then each rule of tree grammar is constructed as follows:

S	T(s)	N(T(s))	P <sub>1</sub>	...	P <sub>k</sub>
left side of grammar	prod. terminal	no. branches $t(t_m)_m$	pointer		pointer $k$

For example, a grammatic rule as  $V_1 \xrightarrow{t_m}$  is represented.

$$\begin{matrix} t_m \\ \swarrow \quad \searrow \\ V_1 \quad V_2 \quad V_3 \end{matrix}$$

In the grammar table as follows:

S	T(s)	N(T(s))	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	...	P <sub>k</sub>
V <sub>1</sub>	T <sub>m</sub>	3	V <sub>1</sub>	V <sub>2</sub>	V <sub>3</sub>	....	



There are three stacks used in the algorithm STACK 1 for input sentence, STACK 2 for grammar table, and STACK 3 for backtrack parsing in the error correction mechanism. The algorithm of the top-down tree parser with error correction ability is as follows:

Algorithm TPEC: (Tree Parser with Error Correction)

Input: Tree grammar  $G_t = (V, r, P, S)$  and tree language of input window

Output: Error corrected sentence.

Algorithm:

- (1) If  $TERMINAL \neq \phi$  go to step 16
- (2) If  $FLAG(TERMINAL) = 0$  go to step 16  
(FLAG equals no. of branches)
- (3)  $Q \leftarrow TERMINAL$
- (4)  $STACK\ 3 \leftarrow Q$ ,  $STACK\ 1 \leftarrow LINK\ 3(Q)$ ,  
 $STACK\ 1 \leftarrow LINK\ 2(Q)$   $STACK\ 1 \leftarrow LINK\ 1(Q)$ .  
(LINK 3, LINK 2, LINK 1, are the three pointers of the tree structure of the input pattern).
- (5)  $\forall$  1 satisfies  $FLAG(Q) = N(T(S(1)))$  where  
 $Q = (T(S(1)))$  go through following steps  
6 - 11 until a successful parse.
- (6)  $STACK\ 2 \leftarrow P_3(T(S(1)))$ ,  $STACK\ 2 \leftarrow P_2$   
 $(T(S(1)))$   $STACK\ 2 \leftarrow P_1(T(S(1)))$
- (7) If STACK 1 is empty, go to 17
- (8)  $Q \leftarrow STACK\ 1$ ,  $FLAG(Q) = K$
- (9) If STACK 2 is empty, go to step 16
- (10)  $R \leftarrow STACK\ 2$
- (11)  $\forall T(R)$ , If  $T(k) = Q$  and all "I" have not  
been tested go to step 5
- (12)  $\forall T(R)$ , If  $T(R) \neq Q$ , then  $Q \leftarrow STACK\ 3$ ,  
 $FLAG(Q) = K-1$  go to step 4, parse the  
combinations of selection  $K-1$  branches  
until a successful parse achieved then  
go to step 13.
- (13) If  $LINK\ 3(Q) = LINK\ 2(Q) = LINK\ 1(Q) = \phi$ ,  
then go to step 7 ( $\phi$  end marker for  
image window edge).
- (14) If  $FLAG(Q) = 0$ , then delete  $Q$ ,  
 $Q \leftarrow STACK\ 3$ ,  $FLAG(Q) = K-1$  go to step 4.
- (15) go to step 4
- (16) reject input sentence
- (17) error corrected sentence (pattern)  
achieved.

## OBJECT DETECTION BY A SYNTACTIC METHOD

The syntactic method for object detection consists of the primitive extraction process and syntactic analysis. The primitive extraction process consists of region primitive extraction and boundary primitive extraction. The primitive extraction process is similar to the one in the syntactic image segmentation algorithm. If the image is not rich in texture, the region primitive extraction process just measure the mean vectors of the image instead of texture features. In this case the computer processing is extremely fast. For the tactical target detection, the VARIABILITY texture measurement of section III is still to be the technique.

The syntactic analysis requires the inference of the tree grammar which generates the boundaries of the objects of interest. The primitive extracted image is processed by the tree parser which is constructed by the set of inferred tree grammars. Thus the object is detected from the scene.

## EXPERIMENTAL RESULTS

The syntactic methods for image segmentation and object detection have been implemented on the IBM 360/67 multi-user time sharing computer at the Laboratory for Applications of Remotely Sensing (LARS). The experiments have been conducted on different LANDSAT, aerophotographic, and infrared images.

1. LANDSAT images. Fig. 1(a) is a LANDSAT image over the Indiana area. The syntactic image segmentation result is shown in Fig. 1(b). The area is 88x88 image. This area has been classified by maximum-likelihood point by point classifier. The comparison of segmentation result and classification result [4], shows that the syntactic image segmentation is quite successful. Also, the computer processing time of the syntactic method is only 63 seconds. But the classification technique takes much longer CPU time than the competitor.

2. Aerophotographic images. For the purpose of showing that this method also works for aerophotographic images, the experiments on aerophotographic images have also been conducted. The image is Fig. 2(a). The image segmentation result by syntactic method is shown in Fig. 2(b).

3. Infrared images. The object detection of army vehicles by syntactic method has been implemented and the experiments on infrared images has been conducted. Fig. 3(a) is the infrared image of a battle field scene. Fig. 3(b) is the object detection result by the syntactic method. The object is successfully detected which is a truck. Fig. 4(a) is another infrared image. Fig. 4(b) is the object detection result and the target is also detected.

## CONCLUSIONS AND REMARKS

In summary, the experiments have been conducted on different images obtained from satellite and aircrafts. The results of syntactic image segmentation compares favorably accurate as those from statistical classification techniques. Also another advantage is that the computer processing time of the syntactic method is much less than that associated with the classification technique.

It has been observed from simulation and experiments that the proposed syntactic technique segments the small areas of textural areas as well as larger ones in the image. The syntactic method for object detection contributes to the military reconnaissance, biomedical diagnosis, and industrial automation. Accompanied with the Syntax-Directed Method [7], this technique contributes to the automation of image understanding.

## REFERENCES

1. K.S. Fu, Syntactic Methods in Pattern Recognition, Academic Press, 1974.
2. K.S. Fu and T.L. Booth, "Grammatical Inference: Introduction and Survey - Part I and II," IEEE Trans. on System, Man and Cybernetics, vol. SMC-5, 1975.
3. K.S. Fu and B.K. Bhargava, "Tree Systems for Syntactic Pattern Recognition," IEEE Trans. on Computers, vol. C-22, No. 12, pp. 262-274, March 1976.
4. J. Keng and K.S. Fu, "Image Segmentation by a Syntactic Method," Quarterly Progress Report Nov. 1, 1976 - Jan. 31, 1977, Research on Image Understanding and Information Extraction, School of Electrical Engineering, Purdue University, W. Lafayette, Indiana.
5. R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," IEEE Trans. System, Man and Cybernetics, vol. SMC-3, pp. 610-621, Nov. 1973.
6. J.S. Weszka, C.R. Dyer and A. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification," IEEE Trans. on Systems, Man and Cybernetics, vol. SMC-6, No. 4, pp. 269-285, April 1976.
7. J. Keng and K.S. Fu, "A Syntax-Directed Method for Land-Use Classification of LANDSAT Images," Proc. of Symposium of Image Science Mathematics, pp. 261-265, Monterey, CA., Nov. 10-12, 1976.

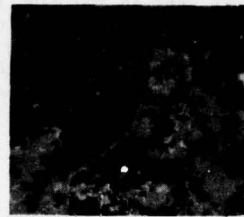


Fig. 1(a) Satellite Image of Indiana area.

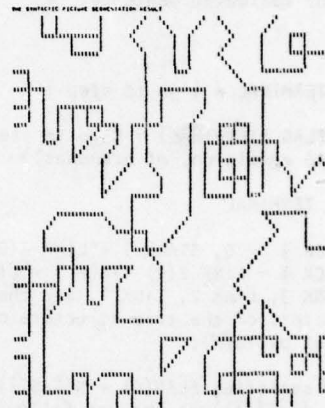


Fig. 1(b) Image segmentation result by syntactic method on Fig. 1(a).



Fig. 2(a) Aerophotographic image of Tippecanoe County, Indiana.

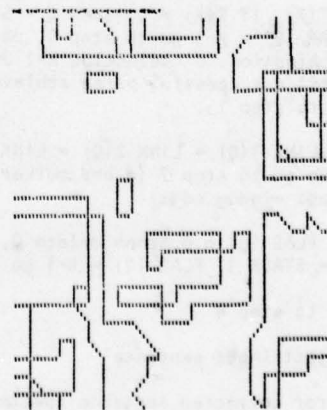


Fig. 2(b) Image segmentation result by syntactic method on Fig. 2(a)



Fig. 3(a) Infrared image of tactical target scene.

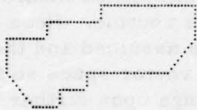


Fig. 3(b) Object detection result by the syntactic method on Fig. 3(a).

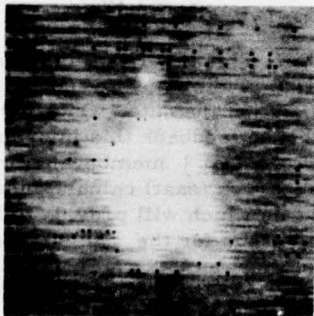


Fig. 4(a) Infrared image of tactical target scene.



Fig. 4(b) Object detection result by the syntactic method on Fig. 4(a).



## A BOTTOM UP IMAGE SEGMENTOR \*

Guy Coleman  
Harry Andrews

Image Processing Institute  
University of Southern California  
Los Angeles, California, 90007

This effort is directed towards a method of automatically segmenting imagery. The method so far developed is autonomous and reasonably fast. A very general block diagram and some preliminary results are shown in figures 1 through 8. In the clustering technique, the only underlying a priori assumption is that homogeneous clusters in N space are desirable; and the search, discovery, and description of these homogeneous clusters is a useful output. Clearly, if these clusters correspond to regions of interest in our imagery, then we have confidence in the possibility of devising discriminating functions for subsequent segmentation. Such techniques for learning of the intrinsic (or lack thereof) homogeneous cluster parameters have come to be known as "unsupervised cluster selection via feature rejection". The numerical processes used in these methodologies are best described as in figure 1. In viewing the figure, certain similarities are immediately obvious with respect to traditional mathematical pattern recognition and supervised learning. Man selects the transducer sensing the image from a vendor, applies his ingenuity to come up with hopefully relevant features and undoubtedly has selected too many such features which are probably correlated and which, therefore, could profit from subsequent decorrelation and feature selection for both equipment minimization and noisy or irrelevant feature rejection.

The next phase of the cluster selection routine has a Bhattacharyya computation which makes one at a time (oat) measurements on each feature (dimension) individually with a feedback input to provide interactive power in the selection process.

\* This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Wright Patterson Air Force Base under Contract F-33615-76-C-1203 ARPA order no. 3119.

We will return to this important aspect shortly. The next box in the figure is the clustering algorithm which is a simplified modification of the traditional Isodata routine. Here a fixed number of clusters, K, is assumed and the data is sorted in N dimensional vector space so that the cluster means do not change upon further sorting under the constraint that the within cluster distance of data points to the cluster mean is minimum. Thus if  $\bar{m}_k$  is the mean of the k<sup>th</sup> cluster and  $\underline{x}_k$  is sorted to belong to the cluster set  $\{C_k\}$ , then

$$\sum_{k=1}^K \sum_{\underline{x}_k \in \{C_k\}} d(\underline{x}_k, \bar{m}_k)$$

is minimum. This algorithm is known to converge to K different cluster mean vectors  $\bar{m}_k$  upon the proper sorting of the data. Once converged, it is now possible to feedback this information, i. e.  $\bar{m}_k$  and  $\underline{x}_k \in \{C_k\}$  membership information to the Bhattacharyya (oat) calculator to compute those features which will provide tighter or better clustering. Thus for the n<sup>th</sup> feature, we have

$$B(C_k, C_j, n) = \log \frac{1}{2} \left( \frac{\sigma_k^2(n)}{\sigma_j^2(n)} + \frac{\sigma_j^2(n)}{\sigma_k^2(n)} + 2 \right) + \frac{\left( \bar{m}_k(n) - \bar{m}_j(n) \right)^2}{\frac{1}{2} \sigma_k^2(n) + \frac{1}{2} \sigma_j^2(n)}$$

for the usefulness of the n<sup>th</sup> feature in separating cluster  $C_k$  from  $C_j$ . For K clusters we have

$$B(n) = \sum_{k > j}^K \sum_{k}^K B(C_k, C_j, n)$$

However, the clustering phase (Isodata) required a priori knowledge of the number of clusters,  $K$ , prior to iteration. Naturally in an unsupervised environment one does not have this information, and as such, one must develop a best cluster definition, which is the objective of the last phase of the algorithm. This phase will feedback the value,  $K$ , to the clustering phase and will monitor the success of subsequent clustering results based upon the number of clusters  $K$  and the feature selection process in trying to define more homogeneous clusters. This final stage bases its decision for the best cluster upon the measure of within cluster scatter (variance) with the between cluster scatter or variance. Let

$$\sigma_w^2(K) = \sum_{k=1}^K \sum_{x_k \in \{C_k\}} (x_k - \bar{m}_k)^2$$

and  $\sigma_b^2(K)$  represents the between cluster variance

$$\sigma_b^2(K) = \sum_{k=1}^K (\bar{m}_k - \bar{m}_0)^2$$

where  $\bar{m}_0$  is the mean of all the data taken as one cluster only. Using  $\sigma_w^2(K)$  and  $\sigma_b^2(K)$  we can compute  $\sigma_w^2(K)$ ,  $\sigma_b^2(K)$  and observe the following property. For  $K = 1$ , (i.e. only 1 cluster), then all data falls into that cluster and there is no between cluster variance. Therefore

$$\sigma_w^2(1) \quad \sigma_b^2(1) = 0$$

At the other extreme when every data point is a cluster itself ( $K = \text{Maximum}$ ) we have no within cluster variation and

$$\sigma_w^2(\text{max}) \quad \sigma_b^2(\text{max}) = 0$$

Because both  $\sigma_w^2$  and  $\sigma_b^2 > 0$  in the range  $1 \leq K \leq \text{max}$ , we know by the mean value theorem that  $\sigma_w^2(K)$ ,  $\sigma_b^2(K)$  must have at least one peak or maximum as a function of  $K$ . And at this maximum we will have a balance between within cluster variation and between cluster variation - Returning to the picture in figure 1 we see that the one image shown is a  $256 \times 256$ , eight bit monochrome image. Features such as brightness and texture are computed at every pixel location in the scene. The output of the feature computation is a  $225 \times 225$  map of vectors where the components of the vectors are the values of the features at the appropriate points in the scene and the size reduction is due to window effects at the scene edges. The next step is to perform a multi-dimensional (Karhunen-Loeve) rotation of the data such that the new features are a linear combination of the old features, but are statistically uncorrelated. This step is performed so that undesirable features

may be discarded and the number of desirable features retained will be the minimum necessary. In other words, the decorrelation prevents retention of several good but highly correlated features.

A preliminary clustering is performed to evaluate the features for their usefulness in segmenting the scene. The evaluation is based on the pairwise average Bhattacharyya distance. Those features which are least useful are discarded and the clustering is performed again.

The clustering algorithm is performed for 2, 3, 4, ... clusters. At each number of clusters, the product of the between and within cluster scatter average is computed. The algorithm is stopped when this product reaches a maximum. The number of clusters and the cluster means are forwarded to the segmentation phase of the algorithm and the image is segmented.

Some preliminary results of this algorithm are shown on the following pages. The segmentations have been subjected to pseudo-coloring to improve the visibility of the different segments.

The first set of pictures resulted from using several variations of the basic procedures on an armored personnel carrier (APC). The first set of APC pictures, called "12 Non-Reduced Correlated Features" is the result of clustering the 12 original features. These features are considered very preliminary and were used to permit development of the clustering algorithm and to verify the ability of the algorithm to reject poor features. The algorithm rejected eight of the 12 features based on the pairwise average Bhattacharyya distance evaluated at the picture labeled "Best Number of Regions".

The data was reclustered, producing the second set of pictures labeled "Best Number of Regions" on the page labeled "4 Reduced Correlated Features" is the end product of the algorithm, having separated the vehicle from the background. The bushes in the top of the scene represent errors, that is, they were classified as being the same as the vehicle.

The next series of APC pictures is labeled "12 Non-Reduced Decorrelated Features". These images are the result of clustering the 12 features produced by the multi-dimensional (Karhunen-Loeve) rotation of the 12 original features. Except for the pseudo-color effects, these images appear quite similar to the "12 Non-Reduced Correlated Features". This is so because rotation of the coordinate axes should not affect clustering.

The pairwise average Bhattacharyya distances for the rotated features were evaluated at the best number of regions (eight in this case) and clustering was performed on the above average features, in this case four. The results of this procedure are shown in the series of images titled "4 Reduced Decorrelated Features". The final result is shown in the image titled "Best Number of Regions," in this case three regions.

The pairwise average Bhattacharyya distances for the 12 rotated features were such that the average for one feature was substantially higher than any of the others. Accordingly, this feature alone was used to perform clustering. The results of this are shown in the final series of images titled "Single Best Decorrelated Feature".

The best number of regions was two in this case. It can be observed that more errors were made in this segmentation than in previous ones due to the enormous reduction in dimension that has taken place.

The second series of pictures is the result of segmenting a color picture of a house. The features used are derived from the red, green and blue color planes of the image and there are a total of 15 (five per color plane). The first picture (two segments) was decided to be the best segmentation based on all 15 features. The additional segmentations are the result of permitting the algorithm to continue segmenting beyond the best number of clusters.



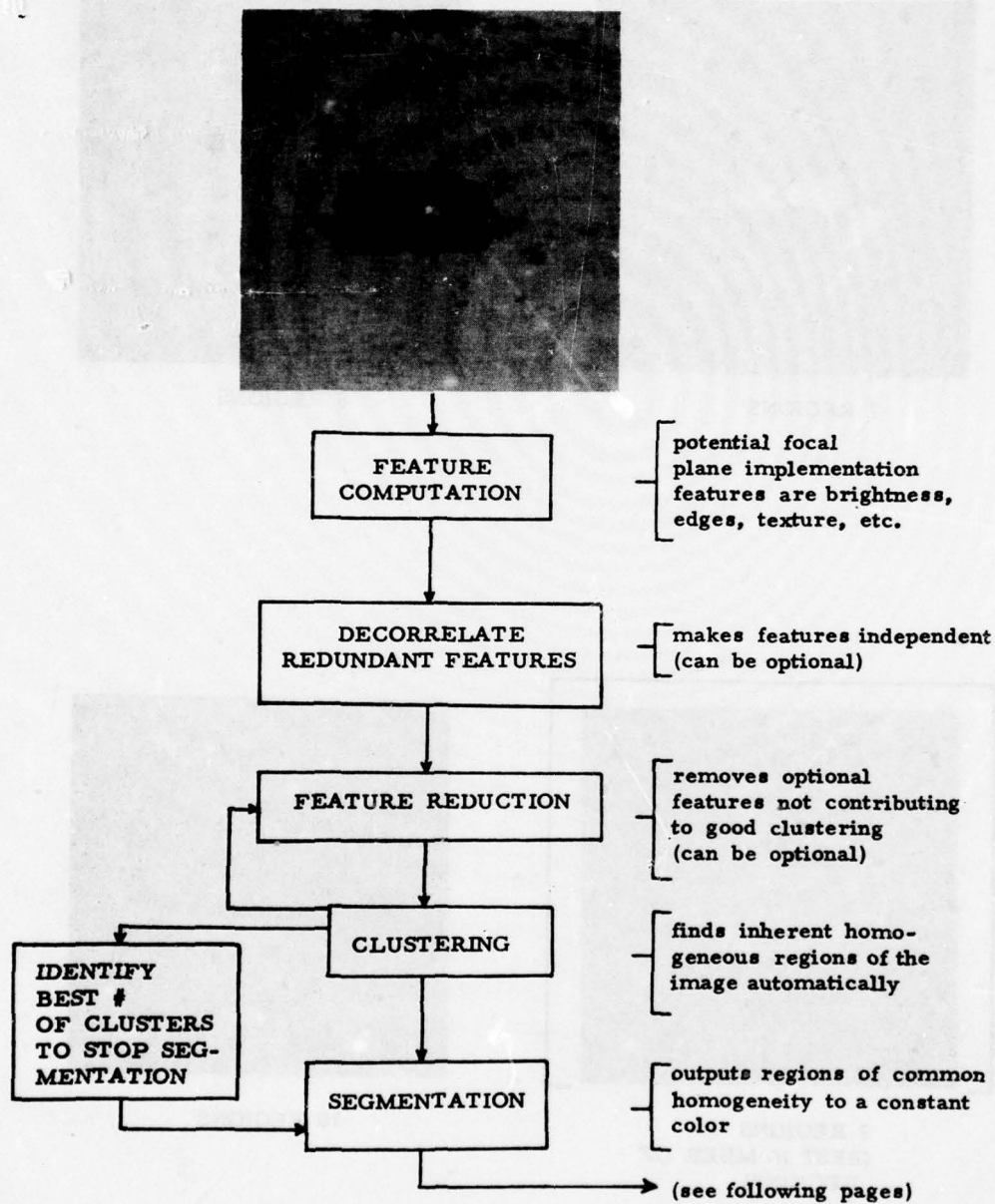
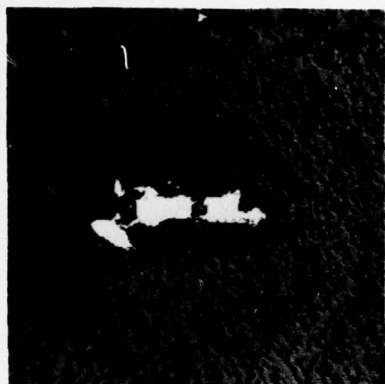


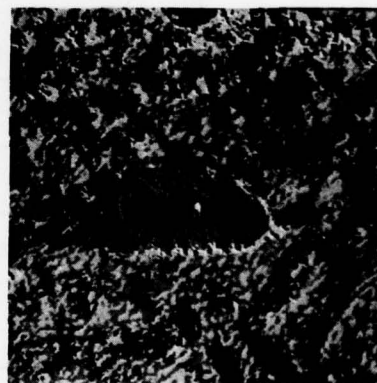
FIGURE 1.



7 REGIONS

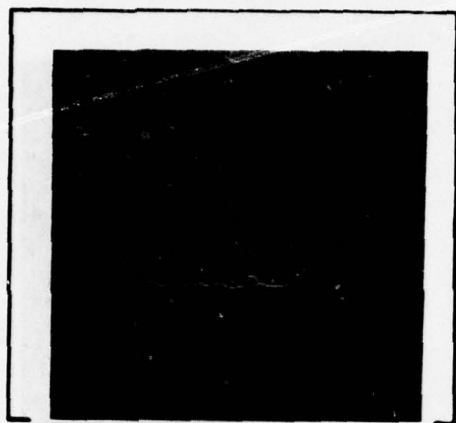


8 REGIONS

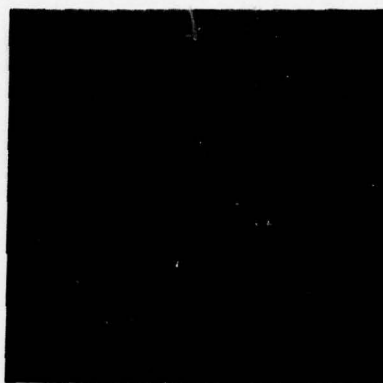
9 REGIONS  
(BEST NUMBER OF  
REGIONS)

10 REGIONS

FIGURE 2. 12 NON REDUCED CORRELATED FEATURES



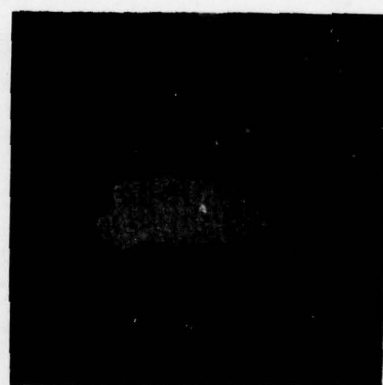
2 REGIONS  
(BEST NUMBER  
OF REGIONS)



3 REGIONS



4 REGIONS



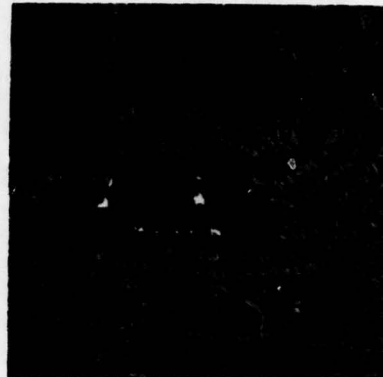
5 REGIONS

FIGURE 3. 4 REDUCED CORRELATED FEATURES

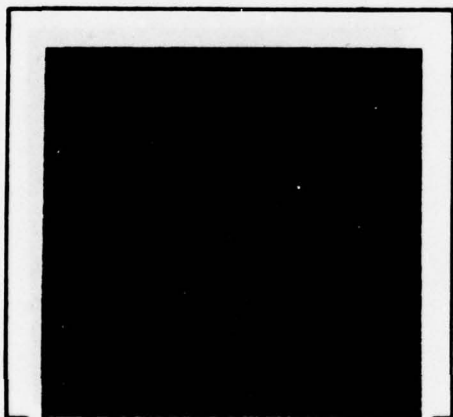




6 REGIONS



7 REGIONS

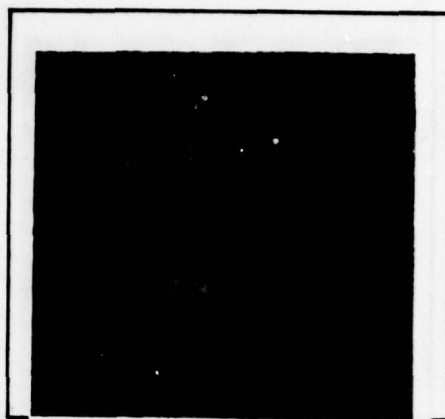


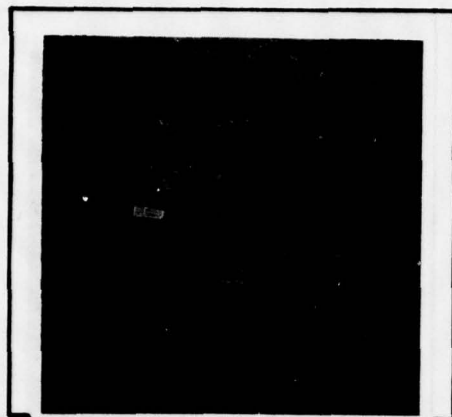
8 REGIONS  
(BEST NUMBER  
OF REGIONS)



9 REGIONS

FIGURE 4. 12 NON REDUCED DECORRELATED FEATURES

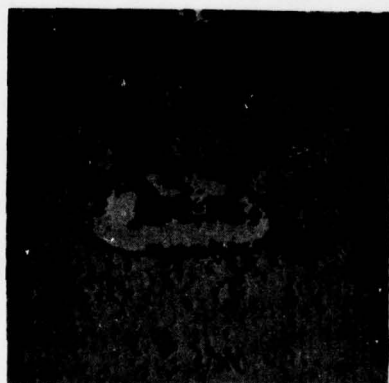
**2 REGIONS****3 REGIONS  
(BEST NUMBER  
OF REGIONS)****4 REGIONS****5 REGIONS****FIGURE 5. 4 REDUCED DECORRELATED FEATURES**



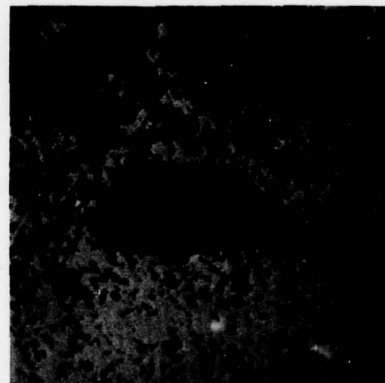
**2 REGIONS  
(BEST NUMBER  
OF REGIONS)**



**3 REGIONS**



**4 REGIONS**



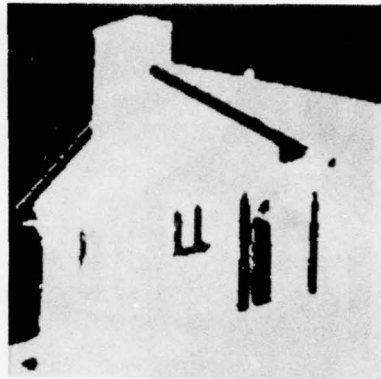
**5 REGIONS**

**FIGURE 6. SINGLE BEST DECORRELATED FEATURE**

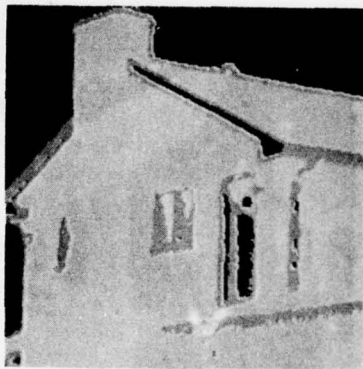




(a) House Original



(b) 2 Regions (Best Number of Regions)



(c) 3 Regions



(d) 4 Regions



(e) 5 Regions



(f) 6 Regions

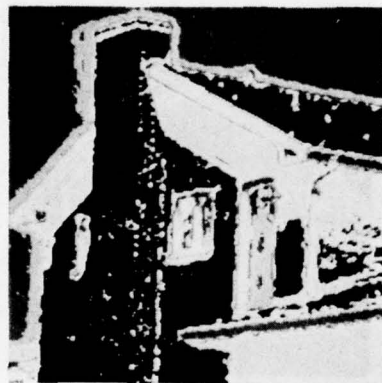
Figure 7. Segmentation of House Picture.



(g) 7 Regions



(h) 8 Regions



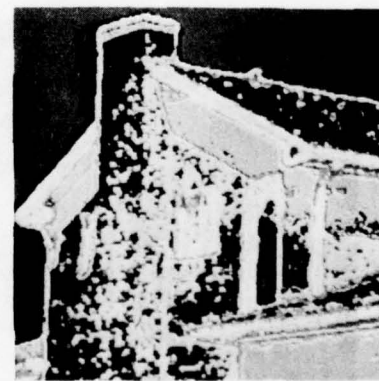
(i) 9 Regions



(j) 10 Regions



(k) 11 Regions



(l) 12 Regions

Figure 8. Segmentation of House Picture.

## A COMPARISON OF SOME SEGMENTATION TECHNIQUES \*

Ramakant Nevatia  
Keith Price

Image Processing Institute and Computer Science Department  
University of Southern California  
Los Angeles, California 90007

### ABSTRACT

Two approaches to image segmentation are edge based or region based. Results using the two types of techniques on pictures of varying complexity, such as a single object viewed at close range and an aerial picture, are presented. Both methods are limited at the current state of development. However, each approach is better suited to particular structures in a picture. Edge techniques are likely to be more suited for extraction of the linear features such as roads, while the region methods perform better for segmentation of the large, uniform, and irregular areas. It is concluded that an image understanding system should exploit the strengths of each to achieve better results than obtainable when each approach is used alone.

### INTRODUCTION

Segmentation is, of course, a key component in the Image Understanding process. The numerous segmentation techniques may be viewed as being either edge based or region based. The edge based techniques start by detection of local discontinuities in some attribute, such as brightness, of an image and attempt to construct object boundaries from them. The region based techniques attempt to find areas in the image over which one or more attributes are constant.

It may be that in some sense the two techniques are trying to compute similar functions and that they should be capable of achieving similar performance. However, at the present state of development of these methods, one or the other technique may be more successful on certain kinds of images. This is a subject of active discussion among researchers in the field, but we are unaware of any comparative studies.

\*This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Wright Patterson Air Force Base under Contract F-33615-76-C-1203.

In this paper, results of processing four selected, black and white pictures using the two classes of techniques are presented that lead to some expected conclusions about their suitability for different tasks. The edge based technique is that developed at University of Southern California [1-2], and the region based technique is that of Ohlander [3], modified by K. Price [4], and developed at Carnegie-Mellon University.

A clustering segmentation scheme that may be viewed as a generalization of the Ohlander technique has been developed by H. Andrews and G. Coleman at USC [5]. However, this technique is in early stages of development and results of this processing are not shown.

### SEGMENTATION TECHNIQUES

A brief review of the segmentation techniques used is provided here.

a) An Edge Based Method - In this method, a local edge operator is applied to an image first. The resulting edges are then linked in straight line segments and only segments of a minimum length or above are preserved (for details see [1]). It is hypothesized that such segments usually correspond to the desired boundaries.

The linking method is independent of the edge operator used. However, the final performance is obviously determined by the output of the local edge operator. We have used a Hueckel edge detector in previous experiments. This edge detector is believed to have superior performance to many simpler edge detectors, but it is not always effective in presence of texture. A simple edge detector, which consists of convolving an image with elongated edge masks in various directions and choosing the maximum was developed and found to perform well (for details, see [2]). This edge detector has been used in results presented later.

b) A Region Based Method - Ohlander segmenter operates by computing histograms of various



image attributes and segmenting the image into regions with a certain range of values of an attribute. The attribute with the best separation (a bimodal distribution in the histogram) is chosen for segmentation. Originally, the method was developed for color images. We have used only black and white images here and only the intensity attribute was used.

This technique is recursively applied to the segmented regions until regions become too small or cannot be further segmented according to established criteria of histogram separations. Regions smaller than a selected size are ignored. Therefore, long thin regions which are broken into several smaller regions may be lost.

#### EXPERIMENTAL RESULTS

The four test images are shown in figures 1(a) through 4(a). Figures 1(b) through 4(b) show the edges detected in the four images. Hueckel edge detector was used for figures 2(b); the edge detector described in section 2 was used for the others. Figures 1(c) through 4(c) show the regions detected by the Ohlander-Price segmenter.

Following are some observations on the relative merits of the two approaches.

(a) The performance on the simpler picture of the truck of figure 1 is comparable. The edge segmentation can be more sensitive, as in separating the back shadow from the truck, but the boundary is fragmented into several segments. Region methods always give closed regions, by definition, which may be easier to handle for some types of objects or processing.

(b) For camouflaged objects, such as shown in figure 2, the region segmentation technique splits the object in many parts. Further, for this example part of the left wing merges with the background region. This part is better separated in the edge picture. Also the outer wing boundaries are detected in spite of the camouflage. These observations may not necessarily apply when the camouflaged object is against a similar background.

(c) In the more complex aerial pictures, the edge technique seems to extract linear features, such as roads, with ease, whereas the region method does the same for parts of image that are homogeneous, for example the lakes in figure 3. Note that the wider roads are extracted as separate regions in both figures 3(c) & 4(c), but that the other roads may only be

indicated by boundaries between regions or not located at all. The edge method detects many of the roads in figure 4 which are not indicated by the regions method.

(d) The more complex parts of the aerial pictures are not adequately analyzed by either technique, for example the lower part of the river or the suburban areas in figure 3. The main difficulty seems to be due to presence of texture and fine detail.

#### CONCLUSIONS

Interestingly, the two methods perform similarly on large areas of the tested images. However, specific structures are handled better with one or the other. The clear implication is that a complete Image Understanding system should utilize both depending on its goals. A straight forward method is to use a specific technique to locate particular types of objects.

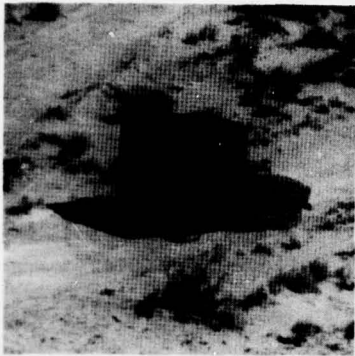
The two segmentation techniques may also be able to reinforce each other at the image level, for example using regions to bridge gaps in boundary segments or to use boundary segments to sub-divide regions. We have not examined such interaction in depth.

#### ACKNOWLEDGEMENTS

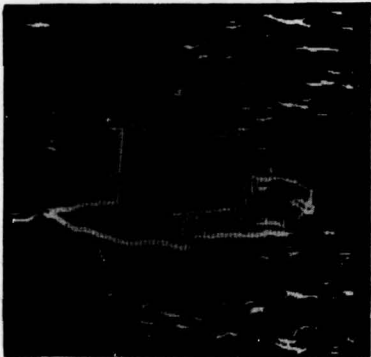
Peter Chuan programmed the edge detector described in section 2 and provided the corresponding results.

#### REFERENCES

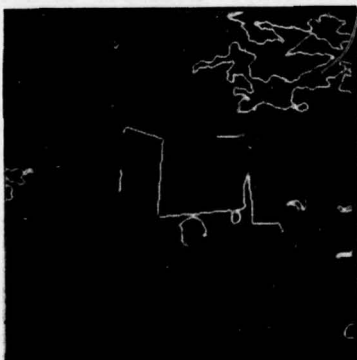
1. R. Nevatia, "Locating Object Boundaries in Textured Environments", *IEEE Transactions on Computers*, Vol. C-25, No. 11, November 1976, pp. 1170-1175.
2. R. Nevatia and P. Chuan, "Detection of Edges in Elongated Neighborhoods", University of Southern California, USCIP Report No. 740, March 1977, pp. 34-39.
3. R. Ohlander, "Analysis of Natural Scene", Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh, Pennsylvania, August 1975.
4. K. Price, "Change Detection and Analysis in Multi-Spectral Images", Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh, Pennsylvania, December 1976.
5. G. Coleman and H. Andrews, "A Bottom Up Image Segmentor", Record of ARPA Image Understanding Workshop, Minneapolis, Minnesota, April 1977. (this volume).



(a) Digitized Image



(b) Detected Edges



(c) Segmented Regions

Figure 1. A Truck



(a) Digitized Image



(b) Detected Edges



(c) Segmented Regions

Figure 2. An Airplane

## REGION EXTRACTION USING CONVERGENT EVIDENCE

D. L. Milgram

Computer Science Ctr., Univ. of Maryland, College Pk, MD20742

## ABSTRACT

Scenes consisting of spatially compact regions which contrast with their backgrounds can be segmented by extracting connected components of above threshold values whose borders match the positions of edges. Edge/border coincidence thus defines a kind of "optimal thresholding", since for any object we can choose the threshold which maximizes the coincidence. This illustrates how the redundancy of different information sources aids segmentation.

## 1. INTRODUCTION

Image segmentation is an important task of scene analysis. When an image has been partitioned into regions, properties of the individual regions can be studied and the regions themselves can be described and, perhaps, identified. Examples where segmentation is important include military target detection, cell classification, and parts inspection.

When the regions to be studied are compact, and correspond to physical entities, they are called "blobs" or "objects". These regions are often characterized by well-defined borders and the contrast of an interior texture with a surrounding texture. Not all scenes conform to this rudimentary model. For example, cloud masses often lack well-defined outlines, and many objects consist of a multitude of subparts with differing textures. Nonetheless, the model is applicable to a variety of diverse imaging environments, including thermal imagery analysis, chromosome classification, and industrial automation.

Thresholding and edge detection as individual aids to segmentation are well described in the literature. The recent book by Rosenfeld and Kak [1] discusses many such segmentation methods. The combined use of interior and edge information has also been investigated. The use of an edge detection operation to suggest a suitable threshold for an image is discussed in [2]. Edge detection was employed by Gupta and Wintz [3] to initialize propaga-

tion processes designed to color in region interiors. Other region growing schemes use high boundary values to guide the merging and splitting of regions defined by gray level similarity. For a survey of region growing, see [4].

One can criticize threshold selection schemes on a number of grounds for attempting to associate a single threshold with a (fixed size) window which, after all, bears no intrinsic relationship to the objects in the scene. First of all, if the window contains no object then attempting to threshold is dangerous, since above-threshold noise regions may often produce probable looking "objects". Secondly, if more than one object is present in the window then a single threshold will not suffice. Thirdly, if an object overlaps several windows then there may be no consistent representation of an object (i.e., no representation using a single threshold). Attempts to divide the scene up into overlapping windows, so that objects of maximal size are guaranteed to lie completely within a single window, answer this last objection at the cost of greatly increased overhead. Chow and Kaneko [5] attempted to overcome these difficulties by assigning thresholds to a coarse grid of points and interpolating threshold values at intermediate points. In any case, the size of the smallest thresholdable region depends on the window size, the coarseness of the grid, and the type of statistical test used to determine if a region is thresholdable. One would prefer, however, to be able to segment a small region regardless of the clutter and noise beyond its borders.

Another objection to pure thresholding is the presence of noise regions in addition to object regions. Noise regions may be difficult to distinguish when based on size, shape or gray level features. The broader and higher the valleys of the gray level histogram, the more likely that the noise regions will be extensive and numerous.

A final objection concerns the design of optimal thresholding techniques



in which the optimality is based on a statistical model of the gray level population. In situations where an object contrasts strongly with the background, there may be a number of thresholds at which the object appears well defined. As the threshold decreases through this acceptable range, each object exemplar is contained within a slightly larger one. Thus although the exemplars may each look reasonable, the optimality criterion for the thresholding does not necessarily choose a "best" exemplar. This is because the optimality condition was based on the whole window rather than on the component corresponding to the object.

For these reasons, we are studying segmentation methods which do not require a commitment to a single threshold in arbitrarily chosen regions of an image.

This paper describes a method for segmenting scenes containing thresholdable objects (i.e., objects that can reasonably be extracted from the image by thresholding). The method uses thresholding as a means of discovering candidate object regions. Candidates are then accepted or rejected based on the coincidence of an edge map with the region boundary. The surviving object regions are compared with the survivors of earlier thresholds, and only those that best match the edge map are used to describe the actual objects in the image. Thus, while a number of thresholds are used, only the one defining the greatest coincidence of thresholded region border and (thinned) edge is deemed valid for a particular region. This method can be considered as defining a best exemplar for each object region.

## 2. METHOD

The basic concept of matching the border points of connected components with corresponding edge values has been mentioned already. However, the implementation of this idea provides an opportunity to vary a number of parameters which can be tuned to respond to different image environments. In this section we present a discussion of the method and its application to a data base of Forward-Looking InfraRed (FLIR) images.

The algorithm may be divided into several steps as follows: image smoothing; extraction of an edge mask by edge detection and thinning; thresholding; forming connected components; and object validity checking. For a given picture, smoothing and edge map extraction need be done only once; whereas thresholding and the subsequent steps are performed over a range of thresholds sufficient to extract any objects in the picture.

Figure 1 illustrates the basic concepts involved. Figure 1a shows several

object windows along with a number of possible thresholds for each. Note that it is not at all obvious which threshold is best. However when the edge map (Figure 1b) is overlaid on the thresholded picture (Figure 1c), we have much better guidance. Figure 1d shows the object region extracted from each window using the method to be described.

### 2.1 Image Smoothing

When random noise affects an image, it has been shown that smoothing can reduce the misclassification error associated with thresholding [6]. Our concern is that misclassified interior points will be interpreted as border points. Since it is unlikely that any significant edge values exist at these points, this would tend to reduce the degree of edge/border point match. Smoothing, by making image regions more homogeneous, improves the classification performance of thresholding.

In our work, we have preferred median filtering to mean filtering as a smoothing operator, since median filtering eliminates small local variations but does not blur edges. The size of the smoothing window is determined not only by the amount of noise that must be eliminated but also by the size of the smallest object to be extracted. A discussion of the tradeoffs involved in median filtering appeared in [7]. A comparison of median filtering with mean filtering for two neighborhood sizes is illustrated in Figure 2.

### 2.2 Edge Detection and Thinning

The smoothed image is the one that is thresholded in later steps. The edge map step may use either the smoothed image or the raw image. The former seems more reasonable since smoothing can be treated as a preprocessing step. In either case, the choice of edge detector is guided by a knowledge of the edge population of the image. A very sensitive edge detector, e.g., the 2x2 Roberts cross gradient, responds to noise edges and can miss slowly rising edges. It is therefore likely to allow many noise component borders to match the noise edges produced. Furthermore, because the Roberts gradient does not respond well to ramplike or fuzzy edges, the border/edge match for true object regions will be low. It is apparent that an edge detector with good noise rejection is needed. This can be achieved by using detectors based on differences of adjacent local averages. The choice of edge detector for the FLIR data base was discussed in [8]. At that time, a detector defined by the maximum absolute value of horizontal and vertical differences of 4x4 averages

was chosen. Since then, a simpler detector which utilizes diagonal differences as well has been designed and is used in this study.

Most edge detectors (with the exception of the Hueckel operator) have appreciable response in the vicinity of an edge. For the purposes of the border/edge match step, the edge response must be thinned to an "edge map". While parallel, iterated thinning methods exist for shrinking narrow regions [9], one can design a one-pass algorithm using non-maximum suppression that utilizes the directional information available from the edge detector. By keeping track of the edge direction (resolved to 45° intervals), the non-maximum suppression can be applied in the direction normal to the edge response, i.e., across the edge. As can be seen, the resulting edge maps provide reasonable "line drawings" of the images. Davis [10] discusses the likelihood that non-maximum suppression will locate the point at which the real-world edge occurs.

### 2.3 Thresholding

The selection of gray levels at which to threshold gives rise to several problems:

- a) The omission of a threshold from consideration increases the probability of missing extractable regions.
- b) The greater the number of thresholds considered, the greater the false alarm rate.
- c) The speed of the algorithm is approximately linear in the number of thresholds used.

The probability of missing an object region due to the omission of a single threshold is the product of the probability that the scene contains an object region and the probability that the object region is discernible (by the algorithm) at exactly the omitted threshold. Although knowledge of the a priori probability is dependent on a model for the scene (which does not at present exist), experiments have demonstrated that an object region which is discernible at all by the algorithm can be extracted over a range of thresholds -- dependent, of course, on the steepness and homogeneity of the edge region bordering the object. Noise regions, on the other hand, do not tend to persist over a range of gray level thresholds. This tradeoff may therefore be posed as follows: By sampling at every  $k$ th gray level, we reduce the workload to a fraction  $(1/k)$  without appreciably increasing the false dismissal rate; however, we lose some redundancy in the extracted data which would help us discriminate object regions from false alarms.

While one may conclude that the false

alarm rate is a function of the input window size, it is more reasonable to use the window size to predict an upper bound. The actual rate is a function of the number of thresholds and the positions of the thresholds in the overall gray level histogram. Experiments [11] in choosing thresholds corresponding to minimum "busyness" indicate that such thresholds generally correspond to valleys in bimodal histograms. If, in general, a maximum busyness threshold corresponds to a histogram mode, and if high busyness at a threshold predicts large numbers of above-threshold components, then one may conclude that certain thresholds are worse than others in producing false alarms -- specifically, those at or adjacent to peaks in the histogram. The quantitative aspect of these assertions is currently under active investigation.

### 2.4 Object Validation

It is well known that the connected components of a binary image can be identified in a single pass. During the pass, many statistics pertaining to each component can be gathered, including area, central moments, shape and gray level features. Of particular interest are features relating to the validity of the components; that is, whether the extracted region really corresponds to an object in the scene. If one considers validity checking to be a classification process, then one can compute a large number of potential features and, using standard techniques, come up with a discriminant function. We have taken the point of view that a good discriminant may be obtained by designating heuristic conditions which an object must satisfy and then assigning one or more features to each heuristic. We have established two heuristics to be of value among the many possible. One is that objects should be "well-defined", i.e., have discernible borders. Note that not all real-world regions satisfy this constraint. For example, in LANDSAT scenes, forests, urban areas and clouds can blend into their surrounds with no discernible edge. The second heuristic is that an object's interior should "contrast" with its surround. In this study, contrast is based on gray level difference. However, other local features including texture measures are worth considering as defining object interior. (This might require the use of texture edge detectors as in [12]). These two heuristics are fairly independent, as will be demonstrated.

The two heuristics just described were embodied in the algorithm as two features. "Well definedness" of a region was measured by the percentage of border points which correspond spatially to (match) actual edge points in the edge



map. "Contrast" was measured by the absolute difference of average gray level between the border region of the component and its interior. Figure 3 shows a scatter plot of these two features for the regions extracted from a set of windows. A reasonable discriminant appears to be:  $\text{match} > .5$  and  $\text{contrast} > .6$  -- i.e., at least 50% of the border matches the edge map, and the contrast is at least .6 gray levels (out of 64). Note that neither feature is by itself reliable enough to discriminate noise regions from object regions. Optimal discriminants may be computed based on several models. Regardless of the particular model chosen, the discriminant value can be interpreted as a "score" for the component. Components with very low scores are discarded as pure noise. In practice, we have used the match measure as a score for objects which were above the pure noise threshold.

The score is important in comparing (nested) object regions corresponding to the same object. When an object is thresholdable at gray levels  $t_1 > t_2 > \dots > t_k$ , this gives rise to  $k$  connected components,  $C_{t_1} \subseteq C_{t_2} \subseteq \dots \subseteq C_{t_k}$ . Since each  $C_{t_i}$  represents the same object, we call each an "exemplar". In general, we wish to select a single exemplar as the best representative of an object. The score provides a criterion for selecting among exemplars. Thus, one could choose the exemplar  $C_{t_j}$

with the highest score. It is not always easy, however, to determine the nested sequence  $\{C_{t_i}\}$ . In particular, if an

object thresholdable at gray level  $t$  is contained within an object thresholdable at gray level  $t' < t$ , then regardless of the comparative differences between the two scores, we would want to retain  $C_t$  and  $C_{t'}$ . This situation can be handled

by assuming that nested components whose areas are sufficiently different (say, 50% change in size) correspond to different (although nested) objects. In thermal images, this might correspond to a warm vehicle with a hot engine compartment, or to a vehicle on an asphalt road. In the first example, the relationship is based on "a part of"; in the second, it is based on occlusion. The results of applying the algorithm to a moderate-size data base are illustrated in Figure 4.

## 2.5 Subsequent Processing

The foregoing algorithmic steps serve as a filter which passes object regions that are deemed to be valid and that correspond to different objects. Each object region can be described variously as a set of points with gray levels, as a sequence of border points, or as a vector

of descriptors. In our work, the goal was to classify the object regions which passed through the filter. However, one can imagine using such regions to create a "map" of a scene to be used for matching or tracking terrain views, for change detection, or for "planning" in the sense of artificial intelligence.

## 2.6 Algorithm

In summary, the algorithm for region extraction consists of the following steps:

1. Smooth the image, if necessary (to promote clean thresholding).
2. Extract a thinned edge picture.
3. Determine a gray level range for thresholding.
4. For each gray level in the range:
  - a. Threshold the smoothed image.
  - b. Label all connected regions of above-threshold points.
  - c. For each connected region:
    - i. Compute the percentage of border points which coincide with significant thinned edge points.
    - ii. Compute the contrast of the region with the background.
    - iii. Classify the region as object/non-object based on the size, edge match and contrast.
5. Construct the canonical tree for the set of object regions based on containment.
6. Prune the containment tree by eliminating adjacent nodes which are too similar.

## 3. CONCLUSIONS

This paper has investigated the problem of image segmentation for scenes consisting of object regions contrasting with a background. We have shown that evidence from multiple sources can be combined to extract object regions while rejecting noise components. The extent to which the different sources conform defines a figure of merit for the region which can be used to select a best exemplar for an object.

## BIBLIOGRAPHY

1. A. Rosenfeld and A. C. Kak, Digital Picture Processing, Academic Press, New York, 1976.
2. J. S. Weszka, R. N. Nagel and A. Rosenfeld, A Threshold Selection



Technique, IEEETC-23, 1974, 1322-1326.

3. J. N. Gupta and P. A. Wintz, Multi-image modelling, TR-EE 74-24, Purdue University, Lafayette, IN, Sept. 1974.
4. S. W. Zucker, Region growing: childhood and adolescence, CGIP 5, 1976, 382-399.
5. C. K. Chow and T. Kaneko, Automatic boundary detection of the left ventricle from cineangiograms, Comput. Biomed. Res. 5, 1972, 388-410.
6. L. S. Davis and A. Rosenfeld, Image smoothing by selective iterative local averaging (in preparation).
7. Algorithms and Hardware Technology for Image Recognition, First Semi-Annual Report, Computer Science Center, Univ. of Maryland, College Pk., MD, October 1976.
8. Algorithms and Hardware Technology for Image Recognition, First Quarterly Report, Computer Science Center, Univ. of Maryland, College Pk., MD, July 1976.
9. R. Stefanelli and A. Rosenfeld, Some parallel thinning algorithms for digital pictures, J.ACM 18, 1971, 255-264.
10. L. S. Davis, An analysis of a simple nonlinear edge detector (in preparation).
11. J. S. Wozzka, Threshold evaluation techniques (in preparation).
12. A. Rosenfeld, A nonlinear edge detection technique, Proc. IEEE 58, 1970, 814-816.

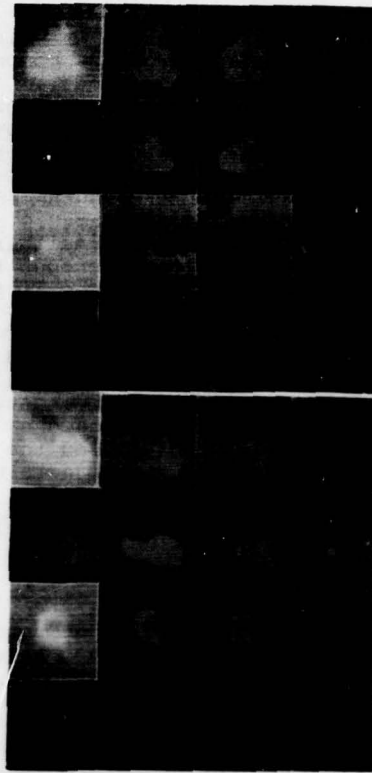


Figure 1a. Four windows (large tank, small tank, truck, APC) thresholded at different gray levels.



Figure 1b. Edge map (thresholded to increase visibility.)

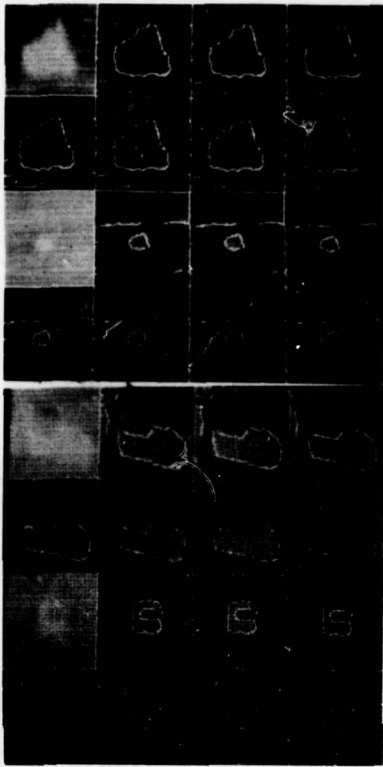


Figure 1c. Edge map of Figure 1b overlaid on Figure 1a.

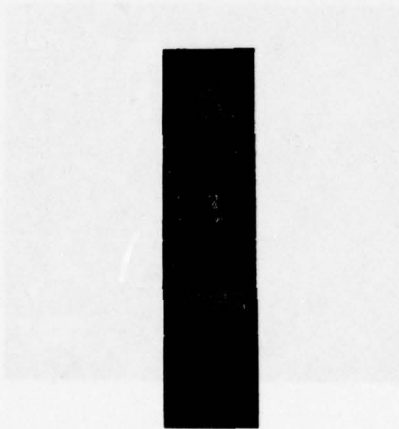


Figure 1d. Object regions extracted by the algorithm.

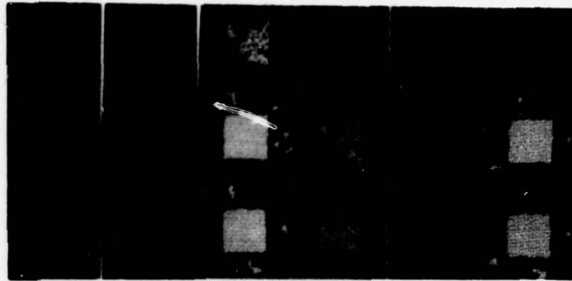


Figure 2a. Noisy square.

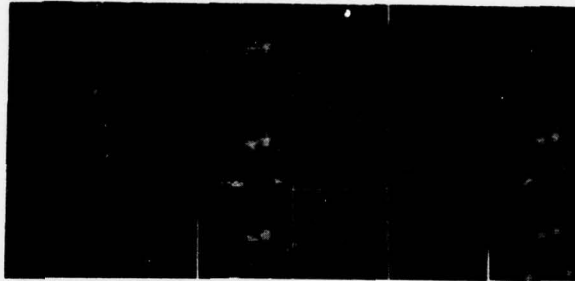


Figure 2b. Noisy tank.

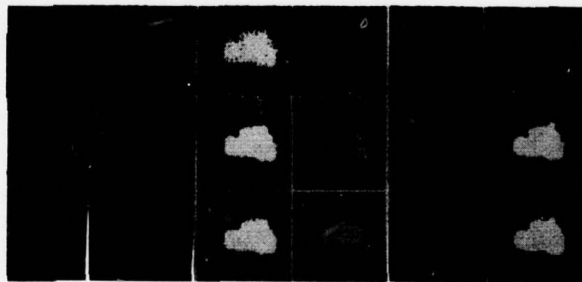


Figure 2c. Large tank.

Figure 2. Object regions produced as a result of image smoothing. (Lighter object regions are displayed within darker object regions.)

- Row 1: Raw window, edge map, object regions.
- Row 2: 3x3 median filtered window, edge map, object regions; 3x3 mean filtered window, edge map, object regions.
- Row 3: 5x5 median filtered window, edge map, object regions; 5x5 mean filtered window, edge map, object regions.

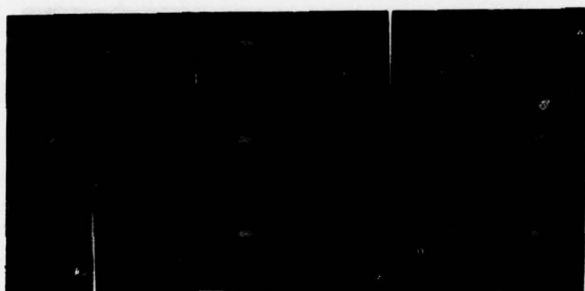


Figure 2d. Small tank.

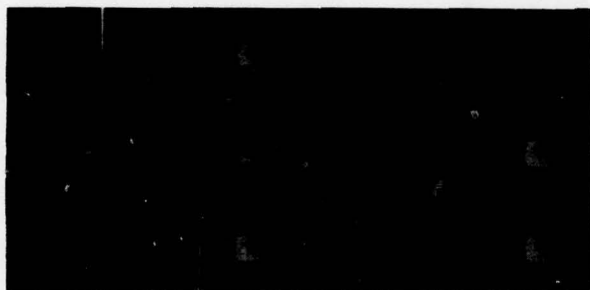


Figure 2e. APC.

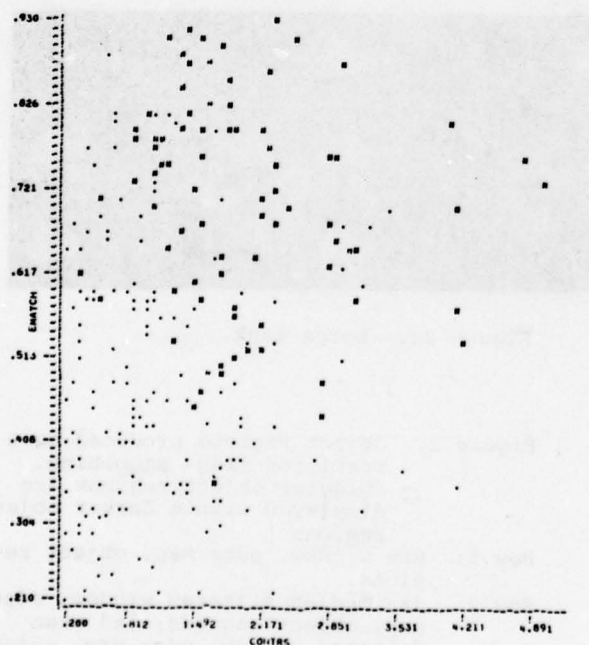


Figure 3. Scatter diagram plotting edge/border match against contrast for a set of noise regions (plotted as periods) and object regions (plotted as hash marks).

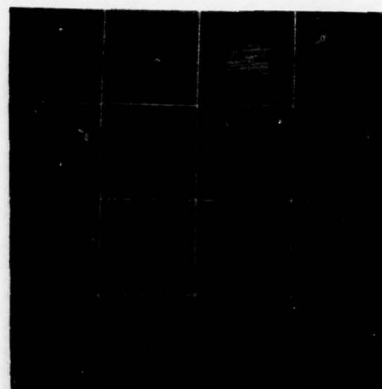


Figure 4a. 16 tanks (the negative frame was not processed).

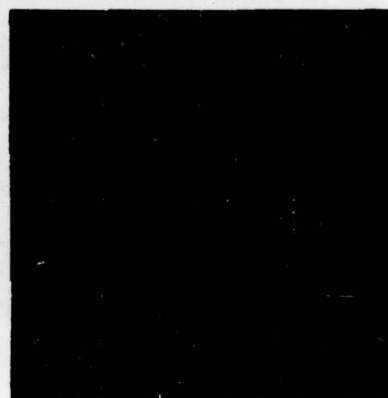


Figure 4b. Edge maps (thresholded for visibility).

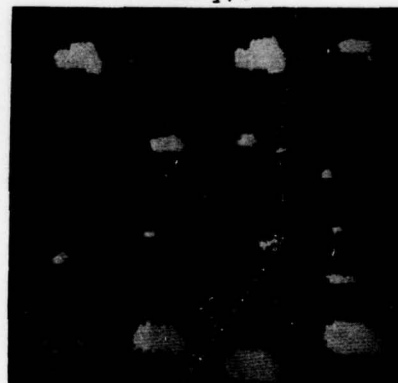


Figure 4c. Object regions.

Figure 4. Object region extraction.



## SEGMENTATION OF FLIR IMAGES BY PIXEL CLASSIFICATION

Durga P. Panda\*

Computer Science Ctr., Univ. of Maryland, College Pk, MD20742

## ABSTRACT

Image segmentation can be treated as a point-wise classification problem. This classification may be done by measuring a set of features at each point and defining a decision surface in the feature space. This report presents some experiments in segmenting FLIR images by using the gray level and the edge value at each point as features.

## 1. INTRODUCTION

Two earlier reports [1, 2] have analyzed the joint histogram of gray level and edge value of FLIR images and have suggested possible segmentation procedures based on the analysis. The analysis indicated that the histogram is trimodal, two of the modes occurring at zero edge value and the third one occurring at some higher edge value and at a gray level between those of the first two. Some of the segmentation procedures suggested in the two reports [1, 2] are: thresholding based on the histogram of gray levels having low edge values; thresholding based on the histogram for high edge values; and valley seeking in the joint histogram. (These methods are defined below.) The present paper investigates the success of these methods in segmenting FLIR images into backgrounds and objects.

The segmentation procedure based on the histogram of points having low edge values (which will be referred to here as the "L-method") finds the valley between the modes in the histogram and uses the location of that valley as the gray level threshold for the image. The segmentation procedure based on the histogram of points having high edge values is referred to as the "H-method"; it uses as threshold the conditional mean or the conditional mode of all pixels with edge value greater than

a certain percentile,  $p$ , of the maximum possible edge value. The quantity  $p$  is taken to be 95; the heuristics leading to this selection are discussed in detail in [1, 2].

Segmentation by valley seeking in the joint histogram involves finding a "bottommost" curve that separates one mode from the others. As described in [1, 2], two such curves are possible. These two curves have a few common points at low edge values and then diverge from each other as edge value increases. Thus, for a given edge value, one of the curves, which we shall call "L", has gray levels smaller than or equal to those on the other curve, which we call "R". Either of these curves may be used as a decision surface in the two-dimensional space of gray level and edge value, for classifying the image points into the object and background classes. The segmentation procedure using valleys such as the curve L or the curve R will be called the "V-method". In particular, the method using the curve R as the decision surface will be called the "VR-method", and the method based on the curve L will be called the "VL-method".

The goal of the work reported here was to investigate the usefulness of the above methods as segmentation procedures for FLIR images, and not necessarily to automate these methods. For this reason the valley selections were done manually.

## 2. EXPERIMENTS

Sixteen windows were selected from the "NVL data base" (see [2]) as test images. The images were 64x64 in size and had grayscale values of 0 to 63. Figure 1a shows the 16 images. The identifiers of these images, as given in [2], are shown in Figure 1b.

Figure 2 shows the gray level histograms of the images in Figure 1. Most of these histograms do not possess strong bimodality. Classically, images can be segmented using thresholds located at valley bottoms on their histograms. In

---

The support of the U. S. Army Night Vision Laboratory under Contract DAAG-53-76C-0138 (ARPA Order 3206) is gratefully acknowledged, as is the help of Mrs. Shelly Rowe.

locating valleys on the histograms of Figure 2, consideration was given to the fact that the object points are much brighter than the background points and they are a small fraction of the total number of points in the image. Thus, a valley near the middle of the grayscale range occupied by the histogram was given less weight, or discarded as spurious, compared to a valley much less prominent but occurring near the light end of the grayscale range. The results of segmenting the images of Figure 1 using valleys found in this way on the histograms in Figure 2 are shown in Figure 3. As this figure shows, only a few of the objects are extracted satisfactorily.

Figure 4 shows the joint (gray level, edge value) histograms of the same images. The horizontal direction represents gray level and the vertical direction represents edge value. The brightness at each point in a joint histogram represents the number of pixels having the corresponding gray level and edge value. The edge operator used is the 4x4 DIFF operator of Hayes and Rosenfeld [3]. The general structure of these joint histograms indicates that for low edge values there is a clearly distinct valley (dark region) separating a strong mode, corresponding to the background region, from a weak mode, corresponding to the object region.

Figure 5 shows the results of segmentation by the L-method, in which the gray level at the valley of the histogram for edge value zero (a common point of the curves L or R) is chosen as a threshold. This method is more successful at extracting the objects, but images with faint objects have very small extracted segments. In image 26R the histogram for zero edge value had no valley, so that the L-method yielded no extracted segment for this image.

Figure 6 shows the result of segmenting the test images by the H-method, as was done in [2]. For many of the test images the objects are well segmented by this method. However, for the images with extremely faint objects the output of the H-method is very noisy. The most undesirable results occur for the two images 38N and 56N, where even though the images contain no object, the H-method classifies some regions as objects. This is due to the fact that the threshold found by the H-method is always within the grayscale occupied by the image and hence it will always yield some segmented regions in the image regardless of whether or not the image contains an object.

Figure 7a shows the test images segmented by the VR-method, while Figure 7b shows similar results for the VL-method. It may be pointed out here that the V-method of segmentation is based on the

concept that an image may contain pixels belonging to three different classes, the background, the object, and the object boundary (see [2]). Pixels belonging to the object boundary class are expected to surround, in the image domain, the pixels belonging to the object class; and are expected to have higher edge values than those of the object or the background pixels, in general. Since the curve R separates the object pixels from both the background pixels and boundary pixels, using the curve R as the decision surface will exclude the boundary points from the segmented image. Comparison of Figure 6 with Figure 7a indicates that this is indeed true in general. However, for the images containing very small and faint objects, such as 57T, 58R, and 34A, the VR-method yields relatively noise-free segments as compared to the H-method. Also impressive is the result of the VR-method for the last two non-target images, 38N and 56N, where the extracted segments are empty. The two-dimensional histograms of these images display no valley and hence, in contrast with the H-method, the extracted segments are empty. Conversely, the curve L separates the background pixels from both the objects and the object boundaries. Thus using the curve L as the decision surface will include in the extracted segments the boundary points that the VR-method excluded. As Figure 7b shows, the extracted segments are larger in the case of the VL-method than they are in the case of the VR-method.

Figure 8 shows the result of using a hybrid of the VR-method, the VL-method, and the H-method as the segmentation procedure. This method classifies the pixels using a decision surface constructed as follows. For a given edge value, beginning with the edge value zero, if the two curves L and R have a common gray level then that (gray level, edge value) pair is selected as a point on the decision surface. As the edge value is increased, the two curves will begin to depart from each other at some point. For this and all higher edge values, the threshold used is the same as the threshold of the H-method. In other words, for low edge values the points on the decision surface are chosen by the V-method, and for higher edge values the points are chosen by the H-method. Some of the extracted segments that were very small in the VL-method are relatively large in the VH-method.

An alternative to the VH-method would be to classify the pixels by the straight line S joining the threshold due to the H-method at the 95th percentile edge value with the threshold due to the L-method at zero edge value. Figure 9 shows the results of segmenting the FLIR images by this S-method. This method does not follow the actual valley bottom



for the low edge values, and the results are therefore somewhat inferior to those obtained using the VH-method. The VH-method yields the best result of all the methods tested in this paper.

### 3. DISCUSSION AND CONCLUSIONS

It is evident that the two-dimensional histogram enables us to extract better objects from FLIR images than those extracted using the one-dimensional gray level histogram. Several different decision surfaces can be used in the two-dimensional feature space, to give varied degrees of success in segmentation. Among all the methods considered, the VH-method seems to give the best results (Figure 8). A heuristic explanation for this is the following. The background and the object pixels away from the object boundary have low edge values, and the histogram for these pixels seems to have a distinct bimodality. Thus the valley between the two modes successfully classifies such pixels into the background class and the object class. The pixels near the boundary, however, have higher edge values and do not have this bimodality. Since some of these points are from the object class and some from the background class, the mean value of such points may be expected to classify the pixels successfully. The VH-method of classification is effectively just that -- the low edge value points on the decision curve are at the valleys of the corresponding histograms, and the high edge value points on the curve are at the mean of the high edge value pixels.

The two-dimensional histograms sometimes resemble truncated or "folded over" mixtures of two multi-variate normal distributions with unequal covariances. The folding over, which occurs at the edge value zero, may be due to the fact that the edge value at each pixel is defined as an absolute value of certain differences measured at that pixel. It is conceivable that if somehow appropriate signs, positive or negative, were incorporated into the edge value at each pixel, the resultant distribution would be an "unfolded" mixture of two multi-variate normal distributions with unequal covariances. In such a case the maximum-likelihood decision surface is quadratic. Unfortunately, how to incorporate the appropriate sign into the edge value at a pixel is not obvious at present.

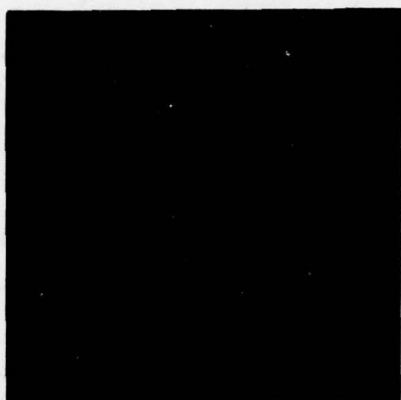
While the use of the edge value as an additional feature has certainly improved the results of pixel classification, it is obvious that the edge value is not the only feature that can be used for this purpose. It is conceivable that there exist other local properties that perform as well as or better than the edge value. Further studies of this approach to image segmen-

tation would be desirable.

### REFERENCES

1. Algorithms and hardware technology for image recognition, Quarterly DARPA Report for the period 1 May - 31 July, 1976, Computer Science Center, University of Maryland, College Park, MD.
2. Algorithms and hardware technology for image recognition, Semi-Annual Report for the period 1 May - 31 October, 1976, Computer Science Center, University of Maryland, College Park, MD.
3. K. C. Hayes, Jr. and A. Rosenfeld, Efficient edge detectors and applications, Tech. Rep. No. 207, Computer Science Center, University of Maryland, College Park, MD, November 1972.





(a)

6T	15T	34T	57T
3R	26R	47R	58R
21A	34A	48A	57A
2N	20N	38N	56N

(b)

Figure 1. The 16 test images.

- (a) The images.  
 (b) The image names. The suffixes T, R, A, and N indicate that the object in the image is a tank, a truck, an APC, or a "non-target", respectively.



Figure 2. The gray level histograms of the test images.

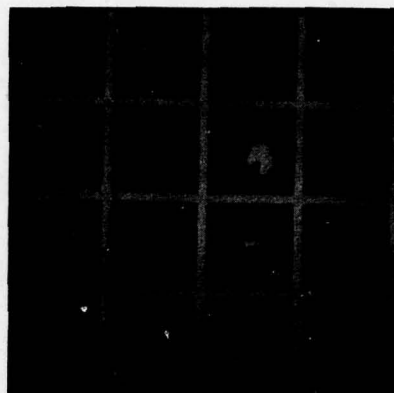


Figure 3. The image segments extracted by finding valleys in the gray level histograms of Figure 2.

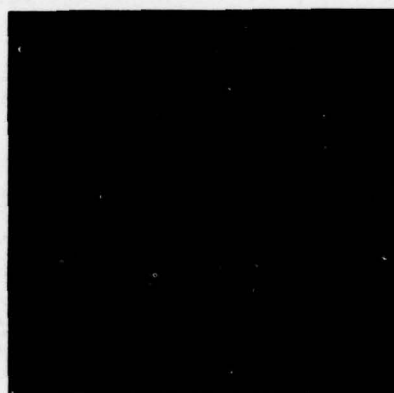


Figure 4. The two-dimensional histograms of the test images.

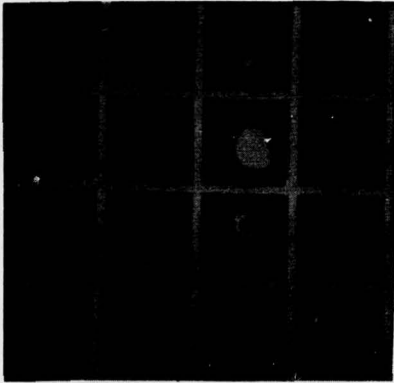
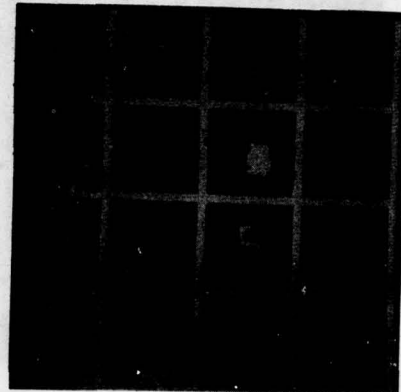


Figure 5. The test images segmented by the L-method.



(a)

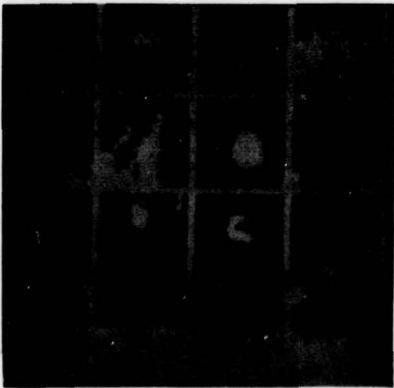
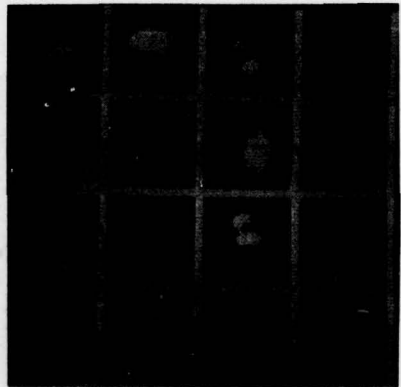


Figure 6. The test images segmented by the H-method.



(b)

Figure 7. The test images segmented by the VR- and the VL-methods.

- a) The VR-method.
- b) The VL-method.

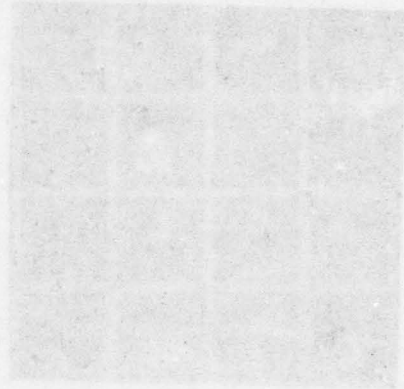
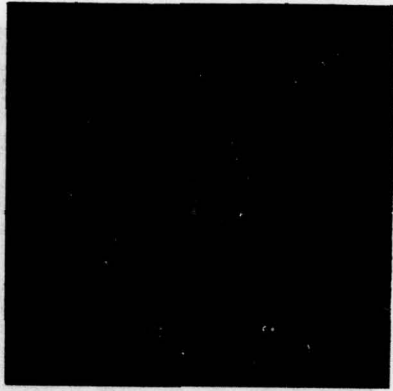


Figure 8. The test images segmented by the VH-method.

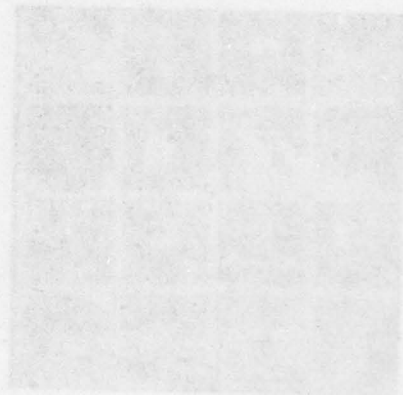
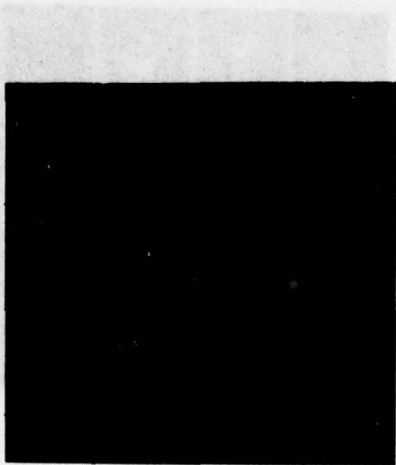


Figure 9. The test images segmented by the S-method.



## IMAGE SEGMENTATION USING TEXTURE AND GRAY LEVEL

S.G. Carlton and O.R. Mitchell  
Purdue University  
W. Lafayette, Indiana 47907

## INTRODUCTION

The segmentation of an image can be a critical first step in image information extraction [1]. Once an image is segmented, each section can be classified using statistical or syntactic methods. However, the segmentation of images has developed into one of the more complex tasks in image processing. This paper presents a hierarchical approach to segmentation using texture and gray level measurements. This method has shown promising preliminary results.

## THE BASIC TEXTURE MEASURE

Several approaches to the use of image texture information in image analysis have recently been developed [2,3]. However, these techniques have generally been applied to region classification following segmentation and not to the segmentation problem itself. In our approach to represent texture, various sizes of local extrema in the logarithm of the image are summed within a window surrounding each point.

## A. Local Extrema

In the work reported here, local extrema were found by combining horizontal and vertical one-dimensional operations. The one-dimensional operation scans a line of data and assigns a point to be a local maximum (minimum) of size  $T$  if it is the largest (smallest) value occurring in the vicinity on the line before the values drop (rise) to an amount  $T$  below (above) this maximum (minimum) value [4]. An example is shown in Fig. 1. The local extrema of size 3 and size 1 are marked. This process is equivalent to detecting the extreme following a hysteresis smoothing operation using a smoothing of  $T/2$ .

The logarithm operation is first performed on the image prior to the extrema detection. The use of this texture measure is a crude attempt to simulate the human visual system's response to a texture pattern. For example, each maximum in Fig. 1 would appear as a bright point to a human observer even though one of them is below a minimum which would appear dark to an observer, i.e., the local surround affects perceived brightness much more than the actual gray level [5].

A sample image is shown in Fig. 2. This is a 256x256 8-bit black and white aerial scene of a military simulation area in New York State. The local extrema measured in the logarithm version of

this image are shown in Fig. 3. Three threshold values are shown as three intensities in the picture (low, medium and high with the extra high omitted). The horizontal and vertical extrema have been combined.

## B. Turning Texture to Gray Level

The next stage in our approach is to count the number of each size extrema in a window centered about each point. This results in a gray level picture representation of a texture property. For example, using a 40x40 window and three thresholds, the three pictures in Figs. 4, 5 and 6 were produced. Note that the forest region of the original has few small extrema, many medium extrema, and quite a few large extrema. The original image was also averaged using the same 40x40 window to produce a fourth picture, shown in Fig. 7, representing the average gray level.

## SEGMENTATION

We now have four pictures (or one 4-dimensional image) to be used for segmentation. Each picture element is considered to be a 4-dimensional vector. To accomplish segmentation, a starting point in each separate segment of the image is found. This is accomplished by finding local extrema in each of the four windowed pictures of section II. In this operation a point must be a local extrema in both the horizontal and vertical directions to be chosen. This prevents the location of starting points in transitional areas between two regions. The starting point candidates are then compared using a four-dimensional distance measure. Each group of similar candidates, based on a threshold criterion, are merged to produce an average vector representing that group. The resulting average vectors form the starting points for the segmentation. The distance measure used indicates an approximate percentage difference in each dimension:

$$D = \frac{4}{\sum_{i=1}^4} \frac{|A_i - B_i|}{A_i + B_i + K}$$

where  $A$  is the intensity of one point in the image and  $B$  is another. This measure is similar to gray level contrast. The constant  $K$  allows for decreasing the weight of a dimension in a region where the total number of extrema is small and, therefore the percentage difference is unreliable. For a window size of 40x40 we used a  $K=25$ .

Once the final set of starting points is determined, each point in the image, regardless of its spatial location, is assigned to the closest starting point using the distance measure described above. This normally results in fairly large contiguous regions due to the nature of the earlier windowing operations. Results using this technique on the image and intermediate steps presented in Figs. 2 through 7 are shown in Figs. 8 and 9. Fig. 8 results when a large distance threshold criterion for similar starting point vectors is used. The additional region shown in Fig. 9 was obtained by tightening this threshold. The major regions extracted from the original image are forest and two different grassy areas.

A simple by-product of this segmentation is the region boundaries. A simple processing procedure on the segmentation output produces the boundary image shown in Fig. 10. These boundaries are then shown overlaid on the original images in Fig. 11.

#### HIERARCHICAL SEGMENTATION

The effects of the window size used in creating the averaged pictures in this procedure are very important. The result of averaging an image with a very large window can be expected to smear all detail from the image. On the other hand, averaging with a small window yields a blurred image retaining much detail and perhaps resulting only in the loss of troublesome noise. In the subsequent proposition these effects of the window size are exploited in a hierarchical approach to the segmentation problem.

Using a comparatively large window 40x40 pixels, the 2 region pictures shown in Fig. 12 were obtained. By examining the averaged pictures and the results, it is readily seen that the fine detail has been lost, but the major regions of the image have been preserved and indicated. This first pass over the image then has produced the major regions.

In the second level of the process, the original extrema are averaged over a smaller window, 20x20 pixels. This smaller window is only 25% of the size of the larger one. The starting points are also regenerated using this newly averaged pictures to form the 4-dimensional vectors. In addition to these inputs, the second level process also utilizes the results of the first level.

The regions indicated by the first level, are separately subdivided by the second level process in the same manner as the first level divided the entire image. The second level process holds the major regions firmly in place, searching within each region for finer detail.

An example of this second level segmentation is shown in Fig. 13. Using the 2 region output of the first level, the right hand region is subdivided into three different regions. The detail preserved at this level is indeed much finer than produced at the first level. The limitations imposed by the first level, however,

provide a spatial context within which the second level may operate. The boundaries obtained are shown on the original image in Fig. 14. The brighter boundaries are those obtained in the first level and those less bright are from the second level process.

#### TECHNIQUE PARAMETER SENSITIVITY

There are several thresholds which must be set to make this technique operative: extrema sizes, window sizes, and distance similarity criteria. However, if the input data is fairly homogeneous (e.g., aerial photographs from a constant altitude) the algorithm performs well using fixed parameters. The algorithm is theoretically invariant to illumination level changes and magnification if the window sizes used are appropriate to the size regions to be detected.

#### ALGORITHM IMPLEMENTATION

The one-dimensional extrema detection algorithm is easily implemented in a line-at-a-time digital processor. The picture is presently transposed and the process repeated to obtain the vertical extrema. It is feasible to implement a two-dimensional version of this algorithm using CCD transversal filter technology which would output extrema sequentially in real time and eliminate the time consuming transposition.

The smoothing operations described are implementable digitally, optically, or with CCD devices. Thus the overall segmentation system could be implemented for very fast image processing rates.

#### SEGMENTATION OF TACTICAL TARGETS IN FLIR IMAGERY

A second type of image data and processing requirement will now be presented. The forward looking infrared (FLIR) imagery as shown in Fig. 15 was obtained from Honeywell as part of our joint project in improving FLIR tactical target detection and recognition. One method of good promise in recognition of such objects involves measuring parameters on projections through the object in various directions. This type of structure recognition method was developed by New Mexico State University for missile tracking at the White Sands Missile Range [6]. It has the advantage that the high amount of noise and distortion present in thermal imagery is reduced by the integrating process of the projections.

Typical projection data is shown in Fig. 16. This shows projections through eight angles of Fig. 15(b). The circles and numbers at the bottom of each projection indicate the locations of intervals containing 10% of the total area above the background level. Ratios of these numbers can then be used in classification of the object. However, this method fails when the background level is comparable or higher than the target intensities. For this reason the target must be segmented from the background before the projections are done.

It is usually a relatively easy task to locate at least one portion of a potential target by looking for intensities significantly different from the background. For active vehicles a motor "hot spot" is usually prominent. The segmentation proposed here assumes that the target has been located and only its extent must be determined.

To accomplish target segmentation, background statistics are gathered over an annular region outside the target region. Then the statistics of the target region are compared to those of the background and points not comparable to those in the background are labeled as target points.

For some targets, intensity level alone can be used as shown in Fig. 17. However, in others it is helpful to include neighborhood variance information as well as gray level or the joint statistics of adjacent points such as was done in Fig. 18. Some targets such as Fig. 15(d), did not respond to any simple statistical measures. In this case the target gray level, variance, and 2nd order density functions were comparable in the background and target. However, the max-min texture method described in this paper did show promise. Shown in Fig. 19 are the vertical extrema associated with Fig. 15(d). Fig. 20 shows the 10x10 average over extrema of medium level. The averaged extrema picture was used in conjunction with the original picture to form a 2-dimensional data vector at each point. The points in the image which now are not common with the background are indicated in Fig. 21. Note that the tank is almost completely segmented.

#### REFERENCES

1. A. Rosenfeld and A. C. Kak, Digital Picture Processing, Academic Press, New York, 1976. See especially Ch. 8 "Segmentation: for a discussion of present techniques and a list of references.
2. J. S. Weszka, C. R. Dyer and A. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification," *IEEE Trans. Syst., Man, Cyber.*, Vol. SMC-6, pp. 269-285, April 1976.
3. R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Trans. Syst., Man, Cyber.*, Vol. SMC-3, pp. 610-621, November 1973.
4. O. R. Mitchell, C. R. Myers, and W. Boyne, "A Max-Min Measure for Image Texture Analysis," *IEEE Trans. on Computers*, in press.
5. T. N. Cornsweet, Visual Perception, Academic Press, New York, 1970.
6. G. M. Flachs, W. E. Thompson and Yee-Hsuun U, "A Real-Time Structural Tracking Algorithm," *NAECON 1976 Record*, pp. 161-168.

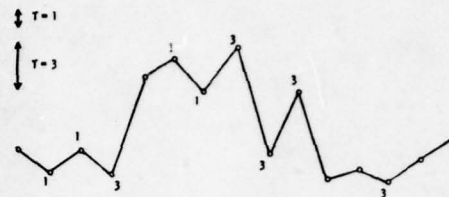


Figure 1. Sample gray level pattern for extrema detection. Local extrema of size 1 and size 3 are indicated.

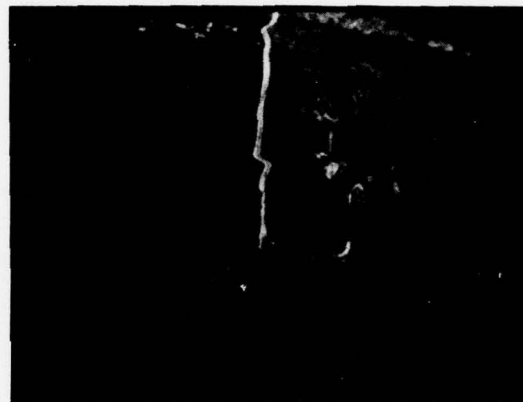


Fig. 2. Original Image to be segmented (256x256) 8 bits.

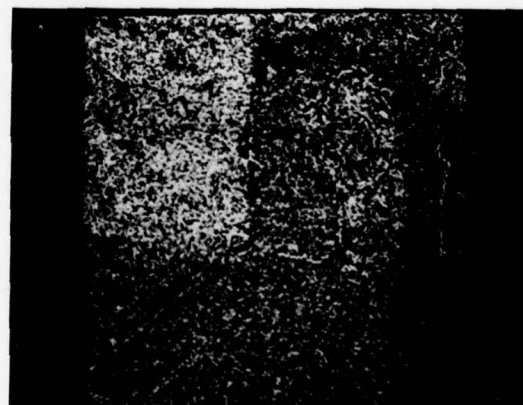


Fig. 3. 3 levels of extrema displayed in different intensities.





Fig. 4. Average low level extrema using a 40x40 window.



Fig. 7. Average gray level using a 40x40 window centered at each point.

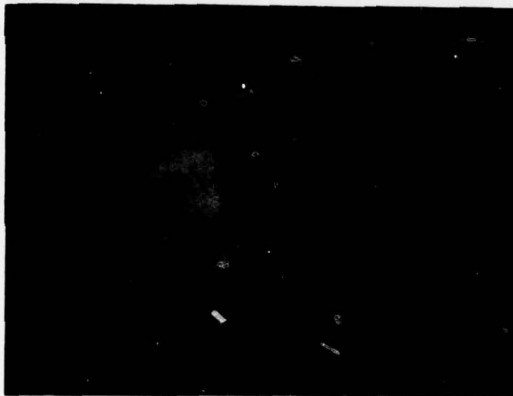


Fig. 5. Average medium level extrema using a 40x40 window centered at each point.

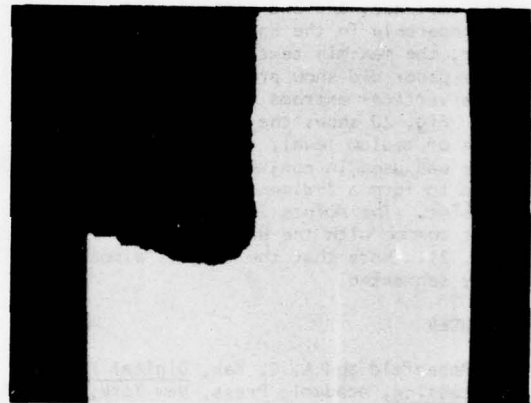


Fig. 8. Results of the segmentation procedure with loose threshold criterion on starting points



Fig. 6. Average high level extrema using a 40x40 window centered at each point.



Fig. 9. Results of the segmentation procedure with a tight criterion on starting point generation.



Fig. 10. Image boundaries produced from the segmentation output.

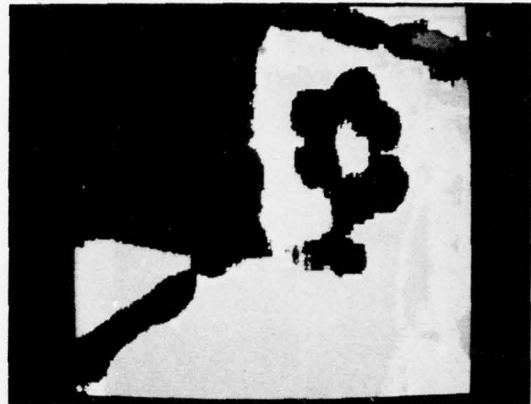


Fig. 13. Output from the second level process showing subdivision of the major region.



Fig. 11. Image boundaries overlaid on the original image.

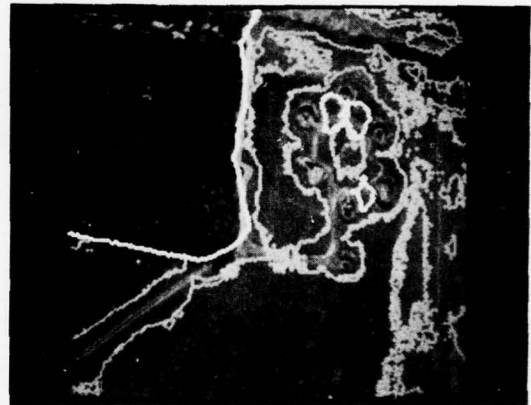


Fig. 14. Results overlaid on the original picture.

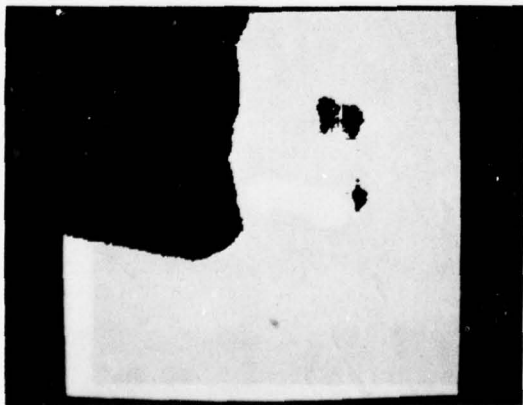


Fig. 12. Two region image used as input to the second level of the hierarchical process.



Fig. 15(a) Typical FLIR data. (91x91) 6-bit picture elements.  
Tank from close range overhead.

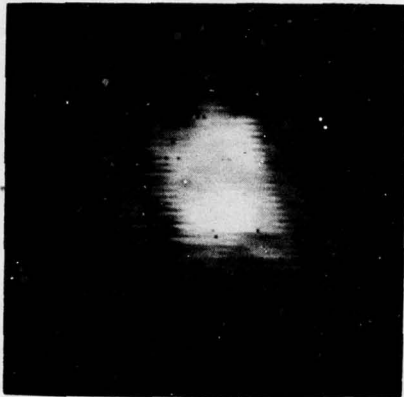


Fig. 15(b) Same tank as in (a) but from further away.

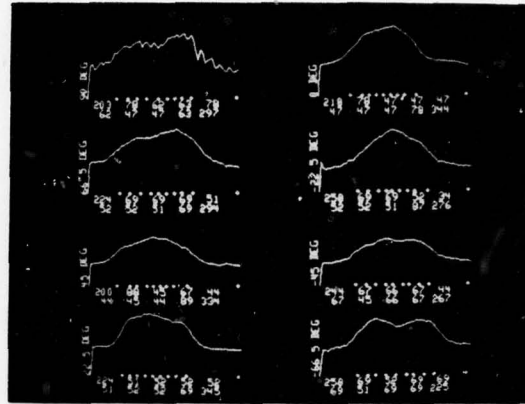


Fig. 16. Projections through 8 angles of Fig. 15(b).

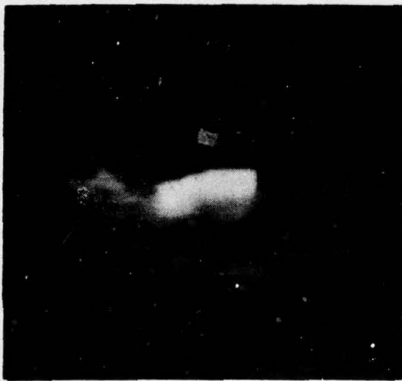


Fig. 15(c) Typical FLIR data (91x91) 6-bit picture elements. Truck

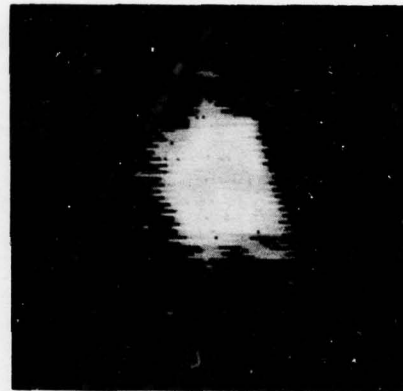


Fig. 17. Segmentation of Fig. 15(b) using background intensity information only.

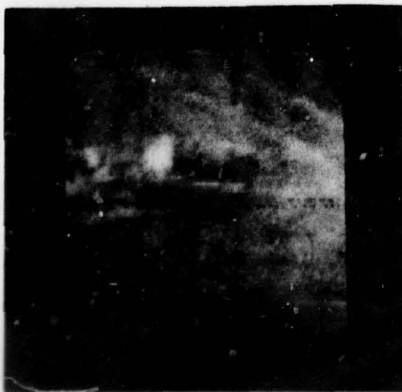


Fig. 15(d) Typical FLIR data (91x91) 6-bit picture elements. Tank



Fig. 18. Target segmentation of Fig. 15(c) using joint statistics of adjacent points in the background.



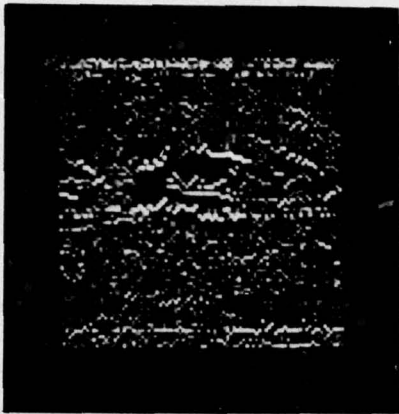


Fig. 19. Vertical max-min extrema for Fig. 15(d).

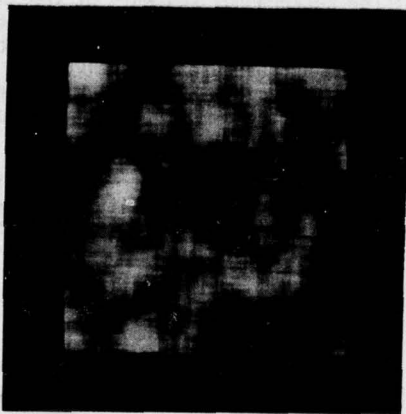


Fig. 20. Average over a 10x10 window of the medium level extrema in Fig. 19.



Fig. 21. Target segmentation of Fig. 15(d) using intensity and texture of the background.

## SYMBOLIC ANALYSIS OF IMAGES USING PROTOTYPE SIMILARITY

R. Touchberry and R. Larson

HONEYWELL INC.  
Systems and Research  
Minneapolis, Minnesota 55413

## ABSTRACT

The prototype similarity transformation is a method for transforming an image into a set of symbols, each of which represents the relationship of a local region to other parts of the image. The relationships that have been investigated are a class of similarity relations invariant under texture changes. Spatial information was not used to determine similarity. Previous work showed that the images were recognizable when the intensities were replaced by the symbols. The current work deals with the problem of simplifying the symbol set in such a way that desired components of the image (e.g., target objects, edges, background) are revealed. A way is given to infer a transformational grammar from the (non-spatial) symbolic content of the image. The purpose of this grammar is to generate a simplified symbol set wherein targets, background areas and separating boundaries are each denoted by a separate symbol. The method has been successful in segmenting FLIR imagery.

## I. INTRODUCTION

Symbolic image processing begins by extracting symbolic primitives from the numeric representation of the image and then working in the symbol space. The goal of symbolic image processing is to determine the content of the image from the structures revealed by the primitives. The problems in symbolic processing are:

- to extract the primitives, and
- to discover appropriate structural relationships among the primitives.

Image primitives are usually defined in terms of required spatial properties of the numeric values. Thus, there is a hierarchy of primitives ranging from edges and contours at the low level to trees and houses at the high level, ordered by the complexity of the spatial properties.

The method discussed in this paper approaches the primitive extraction problem from a different direction. A collection of symbols is derived from the image by means of a similarity relation defined on pairs of image segments. Spatial and textural properties of the image are deliberately excluded from consideration during the symbol generation process. Image primitives are then derived from the symbols, starting with a known target segment and a known non-target segment. Relational graphs are constructed on each of the known segments by

linking similar symbols. The relational graphs yield the simple image primitives

- T - the set of symbols linked only to the target segments,
- B - the set of symbols linked only to the non-target segments, and
- E - the set of symbols linked to both elements

By this association with declared target and non-target (or background) segments of the image, the symbols thus acquire meanings: T = Target and B = Background. The E symbol, since it associates with both target and background is given the meaning: E = edge. Replacing the numerical values in each segment with these symbols gives the transformed image.

The preceding example illustrates the kinds of operations used in the similarity transformation. The significant points, which are expanded in sections II and III are:

- The numerical values (intensities) are used only in testing similarity of pairs of segments.
- The segments are coded to show how they are similar to other segments.
- Semantic cues, in the form of designating a known target segment and a known background segment, are used to guide the final symbol generation.
- The spatial arrangement of the segments does not enter into the symbol generation.

The symbols, which are the image primitives for later analysis of the image content, are thus derived by a process that uses a minimum number of a priori conditions. The utility of the method can be evaluated by how well it helps in determining the content of the image.

The detailed discussion in Section III describes the general procedure that generates symbols representing different kinds of target, background and edge regions in one image. Section IV shows the results of using the prototype similarity transformation to segment FLIR imagery into target, background and edge regions.

## II. PROTOTYPE SIMILARITY TRANSFORMATION

The basic symbols used to represent the image are generated from the image intensity values by the following method. The entire image is partitioned into 4 pel x 4 pel cells. A similarity

relation (a symmetric, reflexive binary relation) is defined for pairs of cells and used to select a set of distinguished cells called prototypes\*. The defining properties of the prototypes are: (1) they are mutually dissimilar and (2) all non-prototype cells are similar to at least one prototype. Each cell is then transformed by replacing the numerical intensities with the list of prototypes that are similar to the cell. The prototype lists are called "labels" and are the basic symbols used by the prototype similarity method. The only intensity characteristics retained in the labels are those used to define the similarity relation, but this is enough information to allow comparing the cells by comparing their labels.

### III. INFERRING THE TRANSFORMATIONAL GRAMMAR

The meaning of the labels is determined by a two step process:

- Infer the meaning of each prototype from the semantic cues provided by the declared prototypes and the collection of labels occurring in the given image.
- Interpret the meaning of each label by using the meaning of its constituent prototypes.

#### Inferring Prototype Meaning

The label of a cell  $C$  is a list of those prototypes that are similar to the cell  $C$ . When two prototypes occur together in a label it indicates a relationship between them. Prototype meaning is inferred by tracing this relationship back to the declared prototypes. For the following discussion let  $P_1$  be the declared target prototype and  $P_2$  be the declared background prototype. Let  $P_3, \dots, P_N$  be the discovered prototypes.

Define  $\Lambda_i$  = the set of labels that contain  $P_i$   
 $L_i$  = the set of prototypes in  $\Lambda_i$ .

$L_i$  is called a "linking" set and is the set of all prototypes related to or linked directly to prototype  $P_i$ . We assume that linking sets containing only one prototype have been removed and that  $P_2 \notin L_1$ . The inference proceeds as follows:

Let  $\pi^1 = L_1 \cup L_2$   
 choose  $P_{j_1} \in \pi^1$ , form  $\pi^2 = \pi^1 \cup L_{j_1}$   
 choose  $P_{j_2} \in \pi^2$ , form  $\pi^3 = \pi^2 \cup L_{j_2}$ , etc.  
 continue until  $\pi^n = \phi$ .

The linking sets  $L_1, L_2, L_{j_1}, \dots, L_{j_n}$  contain the information from which prototype meanings will be inferred.

\*The similarity relation used in this work compares two cells by comparing the ratio of means, the ratio of standard deviations and the correlation coefficient of the ordered intensities to unity.

Index set notation will be used to simplify the following discussion. Let  $J = \{2, j_1, \dots, j_n\}$ . Also define  $L_{ij} = L_i \cap L_j$ ,  $L_{ij}^* = L_i \cap L_j^*$ , etc.

If  $L_{1j} \neq \phi$  then the prototypes in  $L_{1j}$  are like both  $P_1$  and  $P_j$ . These are designated as edges between the target and whatever  $P_j$  may represent. Let  $K \subset J$  be those index values  $j$  for which  $L_{1j} \neq \phi$  and let  $\phi = \bigcup_K L_{1j}$ . Define  $L_1^* = L_1 - \phi$ . The prototypes in  $L_1^*$  are the ones that are like the designated target but are not edges. We designate these as targets.

The prototypes in  $L_{1j}$  are not like the declared target and the next step is to infer meaning for them. Continuing with the idea of using semantic cues to guide the inference process, we consider the declared background prototype  $P_2$  and its linking set  $L_2$ . The subset  $L_{12} \subset L_2$  has already been designated as the target/background edge, leaving only  $L_{12}^*$  to guide the inference. Thus we attempt to find chains of sets  $L_{1j}$  that overlap and join to  $L_{12}$ .

Let  $\Sigma^1 = L_{12}$

choose  $i_1$  such that  $L_{1i_1}^* \cap \Sigma^1 \neq \phi$ ,

form  $\Sigma^2 = \Sigma^1 \cup L_{1i_1}^*$ ,

choose  $i_2$  such that  $L_{1i_2}^* \cap \Sigma^2 \neq \phi$ ,

form  $\Sigma^3 = \Sigma^2 \cup L_{1i_2}^*$ , etc.

continuing until  $\Sigma^r \cap L_{1i}^* = \phi$  for all

remaining  $i$ .

The prototypes in  $\Sigma^r$  are linked to the declared background and we designate these as background. Define  $R = \{2, i_1, \dots, i_r\}$ , then  $\Sigma^r = \bigcup_R L_{1j}^*$ .

At this point three different sets of prototypes have been given meanings:

$L_1^*$  - target

$\phi$  - edges

$\Sigma^r$  - background

If all prototypes are included in these sets then the inference process stops. If not, then meanings are inferred for the remaining prototypes by using the edges  $L_{1j}$  of those  $L_{1j}$  not in  $\Sigma^r$ . Thus:

for  $i \in J - R$ ,



designate  $L_{1i}^-$  as target type i

if  $L_{1i} \cap L_{1r} = \emptyset$  for all  $r \in R$ ,

designate  $L_{1i}^-$  as background type i

if  $L_{1i} \cap L_{1r} \neq \emptyset$  for some  $r \in R$ .

#### Interpreting the Labels

The prototypes have the meanings given them by the sets  $L_1^*$ ,  $\Sigma^r$ , etc. Label meanings are inferred from the prototype meanings by the following process. Each prototype P is replaced by a character denoting the set to which it belongs:

$P \in L_1^*$	T
$P \in \Sigma^r$	B
If: replace P with	
$P \in \emptyset$	E
$P \in L_{1j}^-$	$T_j$ or $B_j$ .

This changes the label into a string of characters. The label's meaning is obtained by using a rewriting rule to reduce the string to a single character. The rule replaces character pairs according to:

TT→T, EE→E, BB→B, TB→E, TE→T, BE→B

ignoring permutations and subscripts. The rule is applied repeatedly until only one character remains.

#### IV. RESULTS

To use the Prototype Similarity (PS) method in a tactical application, a target detection device such as the Honeywell Autoscreener locates the target and points to a location in the target image. A second image location outside of the target is also chosen. These two image points are the semantic cues used in the grammatical inference. The PS method can thus be used to extract the total target sub-image or to find other similar objects in the image. Figures 1 and 2 show the results of using the PS method in these two ways to extract objects from airborne FLIR imagery.

Figures 1a - 1d show the analysis of a truck image. For this analysis the part of the frame used is 60 pels high and 80 pels wide. This is subdivided into non-overlapping 4 pel by 4 pel cells. One target cell and one background cell were declared in the positions shown in Figure 1a and the PS method generates the symbolic representation shown in Figure 1b.\*

\*Note that the direction of the horizontal axis is reversed in Figures 1b, 2c, and 2d.

The details of this analysis are as follows: Sixteen prototypes were discovered in the image. Three of these (shown by the letter Z) were removed because they were not linked to other prototypes. The inference process determined that five symbols (T, O, A, B, E) would describe the picture. The meanings for these symbols (as described in section III) are E - edges between the target and the background, B - background areas, T - areas like the declared target cell, and O, A - target-like areas that differed from the declared target cell.

Figures 1c and 1d respectively show the non-background and the background parts of the picture (the mask is displaced 4 pels vertically for this display). It is noteworthy that the extracted truck image includes the low intensity parts of the image that corresponds to the front and rear wheels.

Referring again to Figure 1b, the declared prototype T was at row 6 and column 13 and corresponds to the right door of the truck. The sides of the truck and top of the cab are the parts of the truck also marked with a T. The symbols O and A were identified as target-like but different from T. These appear in the locations corresponding to the motor and the top of the box. The E symbol is associated mainly with the tandem rear wheels. Future studies will investigate whether target image segmentation of this type is good enough to recognize targets from the symbolic image structure.

Figures 2a - 2d show the analysis of a full image frame (480 x 440 pels). Figure 2a is the original FLIR image taken from an altitude of 3500 feet. The horizon is just above the top of the picture. A number of unidentified far away bright objects appear in the upper half of the picture. The wide light band across the lower part of the picture is a warm river. In front of this are scattered houses (bright) and columns of trees (dark). To analyze this picture, two target prototypes were declared at the positions marked T (on the river) and O (on one of the houses). The background prototype position is marked B. The results of the PS analysis was that a total of 27 prototypes (declared and discovered) were generated and these reduced to four symbols T, O, B, and E. T, O, and B have the declared meanings and E means edges.

Figure 2b shows the picture with all O cells removed (the mask is offset). This symbol not only marks the houses in the foreground and background, but it also delineates a large part of the river's edge. Analysis of the image intensities suggests that this is a result of banding in the house image and could be removed by using a texture measure if so desired. The right hand end of the river is also labelled with the O symbol. This is due to the combined effects of the banding and DC bias errors along the scan lines in the original FLIR output. Image intensity equalization will remove this effect.

Figure 2c shows the locations of the O symbols in a computer listing of the symbolic image. Figure 2d shows the T symbols. These successfully mark the end of the river at which the T prototype was declared, but do not extend into the other end where

the texture and intensity are very different. In its present form the PS method uses no spatial or textural information. As a result the method is sensitive to sensor caused distortions, like the DC bias problem in Figure 2a, that change the quality of the imagery from point to point in the picture. Improvements in sensors and developments in image enhancement methods promise more uniform quality imagery. It is also possible to decrease the sensitivity of the PS method to these distortions by including spatial analysis in the grammatical inference.

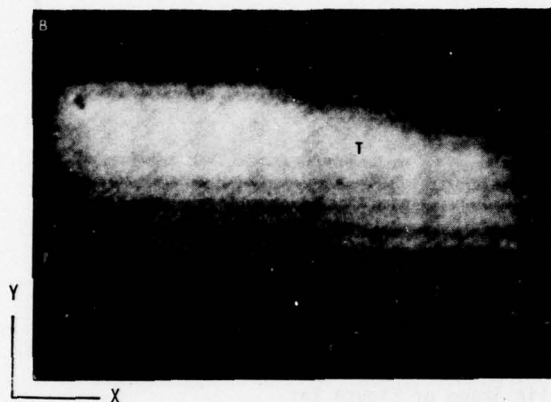


Figure 1a. Original FLIR sub-image of truck.

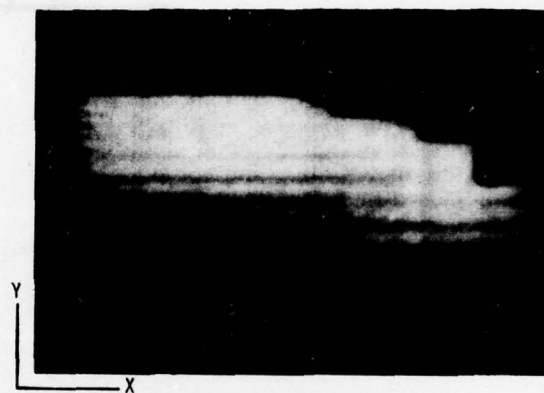


Figure 1c. Target parts of 1a.

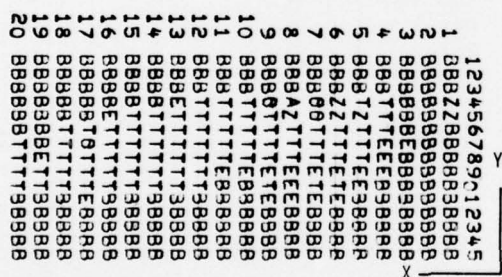


Figure 1b. Symbolic image of truck (B-background; T, A, O - target; E - edges).

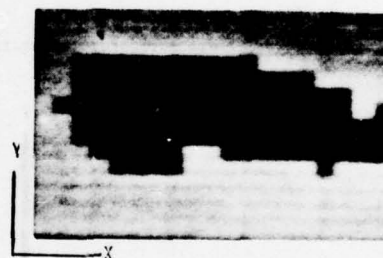


Figure 1d. Background parts of 1a.



Figure 2a. Original FLIR image showing houses, trees, and river.



Figure 2b. FLIR image with building-like cells removed (The mask is offset 12 pels horizontally).

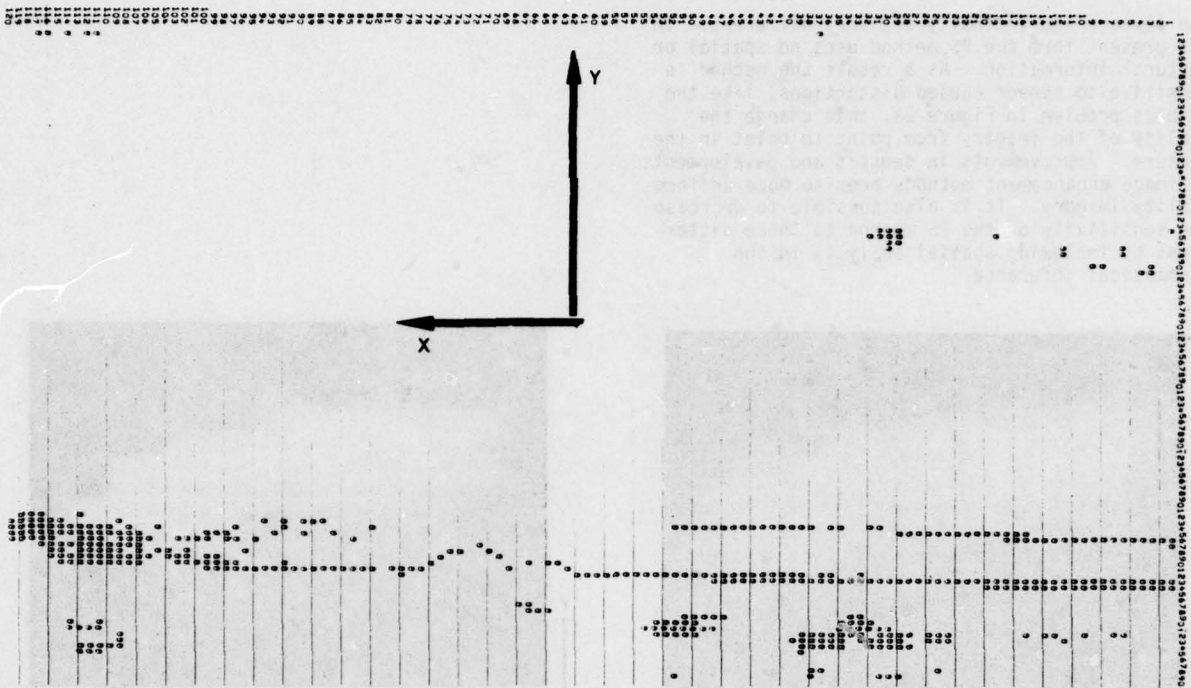


Figure 2c. Locations of building-like cells in the symbolic image of Figure 2a.

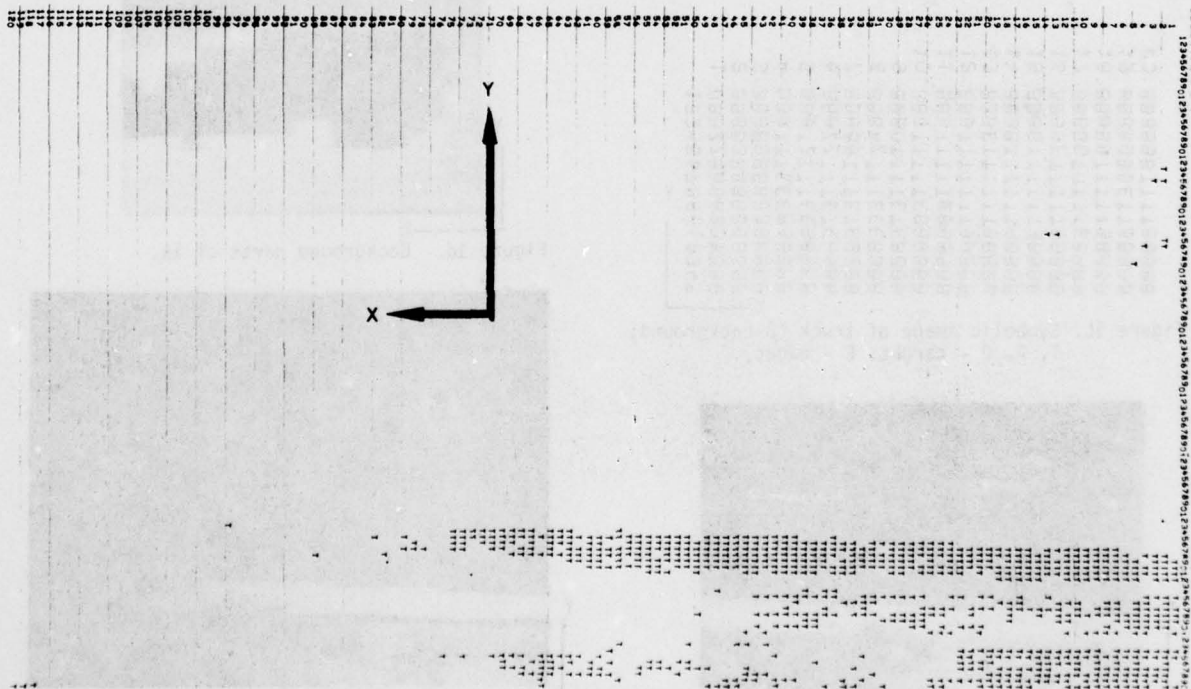


Figure 2d. Locations of river-like cells in the symbolic image of Figure 2a.



## SYSTEM SUPPORT FOR A DISTRIBUTED IMAGE UNDERSTANDING PROGRAM

Jerome A. Feldman and Richard F. Rashid

Computer Science Department  
The University of Rochester

A distributed processing system for image analysis is being developed. Some of the issues under consideration are:

- inter-process communication and flow control;
- programming methodology in a distributed environment;
- effective user control of simultaneous processes;
- protocols for image analysis.

An overview of the effort and its current state will be discussed.

### 1. INTRODUCTION

In the past two years we have mounted a significant effort to develop the tools and expertise necessary for the construction of large distributed systems. This research has taken three directions:

- (1) the design and implementation of an operating system based on inter-process communication (RIG);
- (2) the development of a programming methodology for distributed computing (PLITS);
- (3) the construction of software facilities allowing researchers to manage and interact with a number of simultaneous processes executing on different processors in a network (NEXUS).

These projects have already reached the stage in which they are being routinely used by researchers and students and we have begun to turn our attention to a specific problem domain, that of distributed image analysis.

### 2. RIG--ROCHESTER'S INTELLIGENT GATEWAY

RIG has provided us with a great deal of practical experience in the construction and behavior of distributed systems. The RIG system consists of four 64KW minicomputers (intended primarily for stand-alone use and possessing local

disk storage and high resolution raster displays) connected in a 3 MHz ring network to a Data General Eclipse. The Eclipse maintains a modestly large local file capacity (~100 MB), hard copy printing and plotting, and magnetic tape. It also provides editing and other facilities to a number of local terminals, and serves as a gateway between larger campus machines (360/65 and KL-10), the Arpanet (as a VDH), and our local network.

From the beginning of our project we wanted the central Eclipse to perform three distinct tasks. First of all we wanted to provide our local network of minicomputers with access to the greater storage capacity and I/O capability of the Eclipse. We also needed a consistent communication link between any point in our total system (local minis or campus computers) and any other point. Lastly we wanted to provide users (whether they were connected via terminals to the Eclipse, time-sharing users on the KL-10, or stand-alone users of our minis) with intelligent access to all available facilities, including the various resources of the Arpanet.

To satisfy these goals we designed and implemented a virtual memory operating system called "Aleph" for our central Eclipse. Aleph is based on the concept of inter-process communication. The operating system kernel provides only the necessary framework for interrupt handling, virtual memory management, scheduling and inter-process communication. All other system functions have been allocated to a large number of independent communicating processes each with its own virtual address space.

Aleph's inter-process communication facilities were based on ideas that have been proposed by many (see for example [Walden, 72]), as well as on our own practical experience with the Stanford Hand-Eye system [Feldman & Sproull, 71]. Each Aleph process may have up to 255 full-duplex "ports" for communication with other processes. System primitives allow processes to receive messages from all, one, or a set of ports. When more than one message is waiting on a set of ports, the receiving process may assign priorities which order the messages it will receive. If desired, a process may specify that it will

receive only messages coming from a particular sender. All messages are queued; but a measure of flow control is achieved by restricting the size of a port queue. When a process sends a message to a process-port whose queue is full, that process is either suspended until a place is opened in the queue by the receiver or, optionally, the sender is notified that the system was unable to send the message. A fuller account can be found in [Ball et al., 76].

Of crucial importance to our success in using Aleph as the RIG gateway is the fact that processes do not, in general, share memory or variables. Because of this, they can be easily accessed by other computers on our local network through an aliasing facility which allows external processes to appear local to an Aleph process. We have devised a network communication protocol around the same inter-process communication facilities available in Aleph [Rashid, 76]. Using this protocol, any process on any machine in the system may communicate with processes in the central Eclipse as though it were physically resident. Thus all network machines have full access to Eclipse facilities.

Interestingly enough, because all communication is in the form of messages, there need be no particular importance attached to the physical location of a functional module in the system. Part of the communication protocol includes a symbolic name service which provides processes throughout the network with dynamic information about the location of functional modules in the system. This permits the addition of a second processor to the gateway (an event which should in fact take place within the next month) without causing confusion to the rest of the system. It also means that some of our more prodigious efforts in the area of intelligent resource management, etc. can be developed in very high level languages on our KL-10 and still be tested as an integral part of the RIG system.

## 2. PLITS--THE PROGRAMMING LANGUAGE IN THE SKY

Following the design of RIG and in the light of its ongoing implementation, we began work on a project to develop a non-trivially new programming language for distributed computing. Despite a decade of effort and a massive investment of money and time, there has been relatively little progress in distributed computing. Although many low-level problems have been solved, there is essentially no use of multiple machines on a single task. It is possible that one reason for this is the lack of an appropriate set of conventions for programming a computation that is distributed among several systems.

A description of our current ideas on PLITS can be found in [Feldman, 77]. A preliminary version has been running for almost a year, and is being extended to multiple machines and languages. Our experience with RIG and an initial implementation of PLITS as an extension to the SAIL [VanLehn, 73] language have given us confidence that

it answers at least some of the questions regarding the construction of distributed programs in image analysis and other areas.

## 3. NEXUS--THE PROGRAMMER AT THE HUB OF THE UNIVERSE

One of the goals of RIG was that it give intelligent aid to a user in his access to network resources. As a first step in that direction a powerful gateway should allow the user to manage and interact with a number of network processes simultaneously. NEXUS was devised both as an example of what might be done along these lines and as a useful research tool in its own right.

NEXUS was implemented as a programming environment for our stand-alone minicomputer. It provides inter-process, inter-machine communication in much the same manner as Aleph. More importantly from the user's standpoint, it provides a virtual user input/output system. An invisible process, controllable through special keys, multiplexes user input to the various running processes and the output of these processes can be displayed in various "windows" of the minicomputer's raster display. Over 75 lines can be displayed on a screen, allowing as many as half a dozen reasonably-sized windows to be displayed at a time. Processes running under NEXUS communicate with themselves, with Eclipse processes, and through the Eclipse with the Arpa-net and local KL-10.

Some of the capabilities of NEXUS include Arpa Telnet and FTP, telnet to the campus KL-10, access to Eclipse files and I/O devices, an image display package, and a network file manager. An example of a typical session with NEXUS is given in Figure 1. A similar set of capabilities, although limited by the small (~25) number of lines displayable on a standard CRT, has been implemented for the terminal users of the central Eclipse [Ball et al., 76].

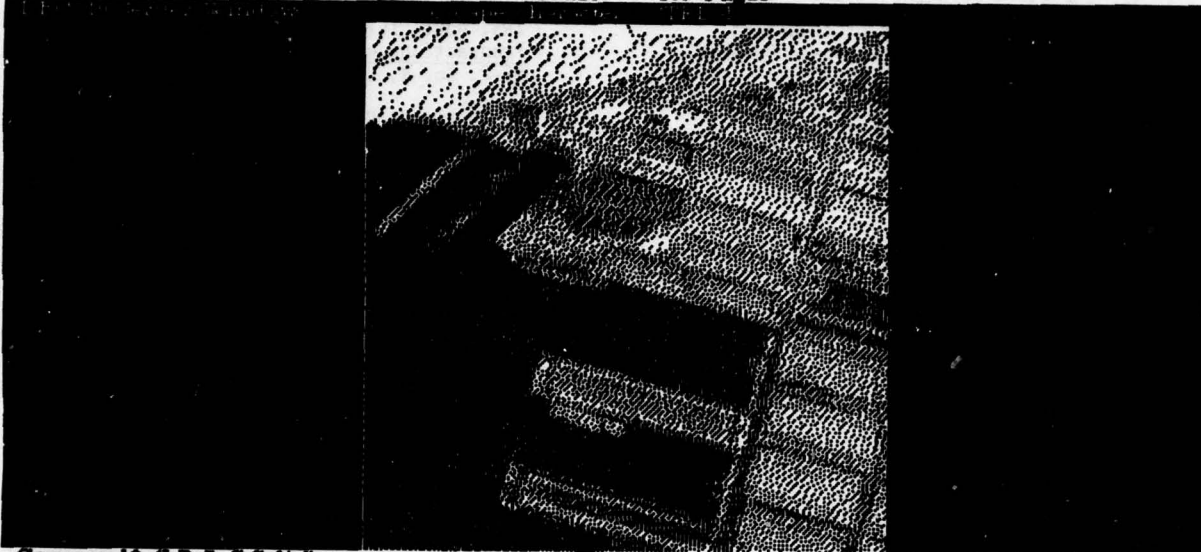
## 4. IMAGE PROTOCOLS

RIG, PLITS, and NEXUS provide some of the system support necessary for distributed image analysis. One way in which we have used these tools is in the development of a network image protocol.

The Rochester Image Protocol [Maleson, Nabelsky and Rashid, 77] exists within the RIG/NEXUS framework and governs communication between image handling processes in our network. It is built around the concept of a structured image definition similar in spirit to the structured graphics display files of [Sproull, 74]. This image data structure serves both as a common language for describing images and as a uniform way of specifying the display of image data on various raster devices (e.g., plotting devices, black & white and color variable intensity and simple intensity displays).

telnet Window  
Type window size in lines -- 7

NEXUS 2.1 -- 4:44:26 -- 159 Pages



Command? (I,P,Q,C,S,?):S  
Show picture.  
Filename:oak1.rv  
Max value:255  
Min value:0  
X length:256  
Y length:256

Command? (I,P,Q,C,S,?):  
#write screen image [ file name]; temp/  
Telnet window: Connection with CMU10B-106  
Use <LF> to enter Telnet command

#open connection with: sumex-AIM  
Connecting...Open <from alto ports> ,1 <to NCP ports> 7,10  
Local Socket=116420  
Foreign socket is 141520000004  
Received CXRULING from port 6; accepting on port 13  
Open on port 13  
Received CXRULING from port 15; accepting on port 15(131)<131>  
SUMEX-AIM Tenex 1.31.77, SUMEX-AIM Exec 1.51.50

@+G  
@log feldman aim  
JOB 17 ON TTY132 28-MAR-77 13:39  
PREVIOUS LOGIN: 28-MAR-77 10:17  
@

and RFP of November 1977  
Sending RFN for ArpaNCP  
Received NRS: Net # Machine #200  
Sending RFC ... Connection Open  
site:

Telnet window: Connection with CMU10B-106  
???034022  
Received CXRULING from port 6; accepting on port 7  
Received CXRULING from port 6; accepting on port 17  
CMU10B 7.T5/DEC 5.06B TTY100 16:44:13  
Type 'HELP' if you need it.  
<255><251>+A



The protocol presupposes the existence of a distributed file system (supported by RIG) and makes use of sophisticated headers to describe the format of image files. Where user interaction (both keyboard and graphic) is required, the protocol also specifies a mechanism for arbitrating between the input requirements of competing processes similar in form to that used by NEXUS.

A full report on the protocol and our initial experience with it is in preparation.

#### REFERENCES

- Ball, E., Feldman, J., Low, J., Rashid, R., and Rovner, P. "RIG, Rochester's Intelligent Gateway: System Overview," Department of Computer Science, University of Rochester, TR5, April 1976. Also appeared in IEEE Transactions on Software Engineering, Vol. SE-2, No. 4, December 1976.
- Feldman, J.A. "A Programming Methodology for Distributed Computing (among other things)," Department of Computer Science, University of Rochester, TR9, 1977.
- Feldman, J.A. and Sproull, R.F. "System Support for the Stanford Hand-Eye System," 2nd International Joint Conference on Artificial Intelligence, London, September 1971.
- Maleson, J., Nabielsky, J. and Rashid, R. "The Rochester Image Protocol," Internal Memo, February 1977.
- Rashid, R. "The Wizard's Guide to RIG: Ethernet Protocols," Internal Memo, October 1976.
- Sproull, R.F. and Thomas, E. "A Network Graphics Protocol," SIGGRAPH-ACM, Vol. 8, No. 3, 1974.
- VanLehn, K.A. "SAIL User Manual," Stanford AI Memo, AIM-204, July 1973.
- Walden, D.C. "A System for Interprocess Communication in a Resource Sharing Computer Network," CACM, Vol. 15, No. 4, 1972, pp. 221-230.

## AUTOMATIC TARGET CUEING ON THE FOCAL PLANE

Thomas J. Willett  
Nathan Bluzer

Westinghouse Systems Development Division, Baltimore

## ABSTRACT

Under contract to University of Maryland, Westinghouse has been implementing algorithms for use in the target cueing process on the focal plane of imaging sensors. The program is sponsored by DARPA, and monitored by the Army's Night Vision Laboratory. It has resulted in an examination of the latest advances in CCD technology and led to the design of innovative structures which require very small chip areas.

We first describe a preferred set of algorithms developed by Maryland which tentatively comprises the first portion of a cueing system. In general, the Median Filter acts to suppress noise. The Gradient Operator extracts edges; the width of these edges is reduced by the Non-Maximum Suppression Algorithm.

The Median Filter is the first algorithm performed and acts to extract the median gray level from a  $5 \times 5$  array of pixels and to place that median value in the center of the  $5 \times 5$ . The median value is defined as the 13th gray level in an ordering of the 25 gray levels by magnitude, counting from either end. The Median Filter acts as a moving  $5 \times 5$  window across the image in that having obtained a median value, the first column is dropped and a sixth column is added with the accompanying reordering to obtain a new median value.

The Gradient Operator Algorithm computes edges based on an image of median values; it computes an operator,  $OP = \max \{ |A-B|, |C-D| \}$  based on four overlapping regions A,B,C,D each of which consists of  $4 \times 4$  pixels and are arranged in the shape of a cross. The quantities A,B,C,D, in the expression, represent the sum of all sixteen pixels within each region. The operator OP works as a moving window in that the leftmost column is removed from each region and the next right column is added to each. Again, the operator result is placed in the center pixel location.

The Gradient Operator extracts edges in either the horizontal or vertical direction; the Non-Maximum Suppression Algorithm then looks in a direction perpendicular to the edge for a larger gradient. If a larger value cannot be found, the edge under consideration is retained; the edge is removed if a larger value is found. The neighborhood around the gradient under consideration is

an approximate  $3 \times 3$  on each side.

The Median Filter, Gradient Operator, and Non Maximum Suppression Algorithms are calculated for a small window which moves over the entire frame. The window height is 5 lines, 8 lines, and 7 lines of image respectively for each of the algorithms which are performed sequentially. Consider, now, how these lines can be obtained from the image.

We assume that the focal plane has a time delay integration (TDI) feature such that the image is available one line at a time from the focal plane. The pixels within each line arrive in parallel and are then shifted out serially into a serpentine delay. For the Median Filter, the serpentine delay comprises 5 image lines. There are non-destructive taps placed strategically in the serpentine such that as the 5 lines are shifted through the serpentine they are tapped to form a  $5 \times 5$  moving window. The same sort of serpentine structure will be used for the Gradient Operator and Non-Maximum Suppression Algorithms with 8 lines and 7 lines respectively. Since the algorithms are performed sequentially, the chip area below the focal plane is composed of a 5 stage serpentine delay, a Median Filter, a 8 stage serpentine delay, a Gradient Operator, a 7 stage serpentine delay, and a Non-Maximum Suppression Algorithm.

It appears that the computation speed of the algorithms will be in the neighborhood of 100 KHZ, hence a parallel organization is necessary for a 1 megapixel/sec. data rate. Suppose we divide the PI/SO register immediately below the focal plane into ten vertical sections each approximately 68 pixels wide and each with its own serpentine CCD delay line. If the image is 640 pixels wide, we divide the register into ten sections of approximately 68 pixels each to avoid problems associated with calculations along the edges. However if we do this segmentation to achieve the data rate, there will be  $20 \times 68$  shifts per column. At a clock frequency of 100 KHZ, numerical degradation in the order of 20% will occur, which is too high.

The modulation transfer function is a function of the input signal frequency, the frequency of shifts (clock frequency), the number of shifts and the transfer efficiency. The more practical avenues of reduction are clock frequency and the number of shifts; we can double

the number of operators to 20 each, and halve the clock frequency and number of shifts to 50 KHZ and 680, respectively. This may produce an improvement to 10% degradation but this number would have to be confirmed experimentally. Of course this approach increases the total chip area which is still small and the external clocking circuitry. Operating at cryogenic temperatures will probably increase the transfer efficiency somewhat. Moreover, surface channel CCD's are suitable for this task and the advantage of these devices is realizing the non-destructive taps. The size required for achieving a serpentine memory 680 elements long is 1000 square mils if four phase clocking is employed. Hence for 20 columns we will require a silicon area 1000 mils long by 20 mils wide.

The size of the Gradient Operator chip will be deduced by assigning real estate to each operation performed by the Operator. A key operation is the absolute subtraction module (ASM) which obtains the absolute difference between two inputs and yields a charge representing that quantity. Each difference CCD structure will nominally require a channel 1.2 mils wide; four input channels are needed to provide four charge packets, two representing  $|A-B|$  and two representing  $|C-D|$ . The length of each ASM will be 4 mils, a size sufficient to provide a read-out structure necessary to drive the second stage of the Operator. The second stage selects which output  $|A-B|$  or  $|C-D|$  is the largest gradient of the  $i$ th pixel location. Combining the real estate requirement for the first and second stages, we calculate a chip size of 8 mils x 10 mils. We assume a four phase gate construction; a smaller number of phases (which requires less chip area) could be used; however, speed - charge handling capacity and ease of fabrication favors four phase construction. The structure advocated is exclusively based on MOS FET and CCD technology. Both MOS FET and CCD structures exhibit improved performance at cryogenic temperatures greater than 30°K. Fabrication yields should be in the neighborhood of 50% and power consumption less than 10 milliwatts.

The Median Filter chip will operate on 25 pixels located within a moving window; provisions for obtaining the 25 pixels will be built into the CCD serpentine delay structure in the form of non-destructive readouts. Each data element (pixel) is assumed to have a dynamic range equivalent to a 32 level gray scale.

The size of the chip is determined primarily by the number of pixels and gray levels. The proposed MFO is required to operate as a moving window device which requires a CCD memory capable of storing and shifting 25 data elements each of which is quantized within a 32 level gray scale. A bank of CCD memory registers with 25 x 32 storage locations can be achieved by a 64 mil by 64 mil module. Included in this estimate are areas for incorporating output and input structures to the CCD memory.

Another major block of the MFO is the sorting module in which the data elements are arranged according to size. This requires a bank of 32 CCD shift registers which are 25 elements long and each row is capable of being independently shifted left or right. An area of 100 mils wide by 64 mils long is sufficient.

Finally the area required for controlling the clocks operating the sorting module is estimated to be 100 mils by 2 mils.

Summing the different component areas comprising the MFO, we arrive at an area estimate of 100 mils x 128 mils.

All the elements used in modeling the MFO are based on field effect phenomena, hence we expect improved performance of cryogenic temperatures in accordance with experimental observations. Power requirements are less than 100 milliwatts at 100 KHZ.

Assuming that the focal plane is divided into 20 columns, the geometric area for the Median Filter, Gradient Operator, and associated serpentine delays, including the delay for the Non-Maximum Suppression Algorithm, is 1 inch x 1/2 inch.



# CCD IMAGE PROCESSING CIRCUITRY\*

Graham R. Nudd

Hughes Research Laboratories, Malibu, California

## ABSTRACT

The rapid development in charge transfer and MOS technology allow highly complex circuit functions to be built in a single integrated circuit capable of operating at speeds in excess of 10 MHz. This paper describes the development of n-channel MOS circuitry for real time (equivalent to TV) implementation of selected algorithms; including edge detection, using the Sobel operator, and unsharp masking. Particular emphasis directed toward establishing the feasibility of performing two dimensional and nonlinear operations in the analog domain and maintaining an accuracy equivalent to 6 to 8 bits. The circuits described typically occupy a few hundred mil<sup>2</sup> on the silicon surface and hence offer great potential for both highly parallel operation and integration with the newly developed imagers.

## I. INTRODUCTION

Until relatively recently the computational complexity of most image processing algorithms prohibited the effective use of integrated circuits (ICs) to process data. However, the rapid progress in technologies such as charge transfer devices and metal oxide semiconductors (MOS) with inherently low power-delay products has resulted in a very significant increase in the circuit complexity permissible in a single IC. (The hand calculator and single-chip micro-processors are commercial examples of these developments.) Charge coupled devices (CCDs) are particularly significant to image processing since they can be employed both in the image detection and processing. Further, they can be configured to provide an especially simple and direct means of performing two dimensional convolutions, which form the basis of much low-level image processing. The Sobel edge detection circuit, described below, is an example of this. Finally, the extremely high packing density of CCD and MOS circuitry allows many circuits to be used in parallel to provide an area processor as shown schematically in Fig. 1. Here a CCD imager or analog store is

used to store a full frame, and the data from the N rows are clocked out in parallel into N parallel processing circuits. Each circuit might perform the Sobel operator, for example, and process the data for an entire line, with the processed output appearing at the clock rate  $f_c$  (which for our circuits could be as high as 10 MHz). Thus, an entire frame would be processed in  $N/f_c$  seconds. For a 512 x 512 frame this would amount to 50  $\mu$ sec. The advantages of these techniques for direct focal plane processing are clear.

We describe here two test circuits to be implemented as n-channel surface CCD's with nominal operating speed of 10 MHz. We are currently fabricating these circuits and designing experiments to evaluate their performance. Finally, we describe the test facilities we have built, based on an Intel 8080 microcomputer to test the concepts.

## II. TEST CIRCUIT I

The first test circuit is a CCD implementation of the Sobel edge detection algorithm. This circuit was chosen because it demonstrates two operations important to image processing; the possibility of achieving a two-dimensional convolution with arbitrary weightings and the ability to perform nonlinear functions such as the absolute magnitude operation.

The algorithm itself operates on an array of 3 by 3 picture elements with intensities  $f(i, j)$  and evaluates

$$S\{i, j\} = \frac{1}{8} \left[ \left| \left( f(i-1, j-1) + 2f(i, j-1) + f(i+1, j-1) \right) - \left( f(i-1, j+1) + 2f(i, j+1) + f(i+1, j+1) \right) \right| + \left| \left( f(i-1, j-1) + 2f(i-1, j) + f(i-1, j+1) \right) - \left( f(i+1, j-1) + 2f(i+1, j) + f(i+1, j+1) \right) \right| \right] \quad (1)$$

for each picture element. A schematic of the circuit concept is shown in Fig. 2. Three parallel lines of charge, proportional to the pixel

\* A more detailed discussion of this work can be found in the USC Semi-Annual Technical Report, dated Sept. 30, 1976.

AD-A052 900

SCIENCE APPLICATIONS INC ARLINGTON VA

F/G 14/5

IMAGE UNDERSTANDING PROCEEDINGS OF A WORKSHOP HELD AT MINNEAPOLIS--ETC(U)

APR 77 L S BAUMANN

F30602-76-C-0165

UNCLASSIFIED

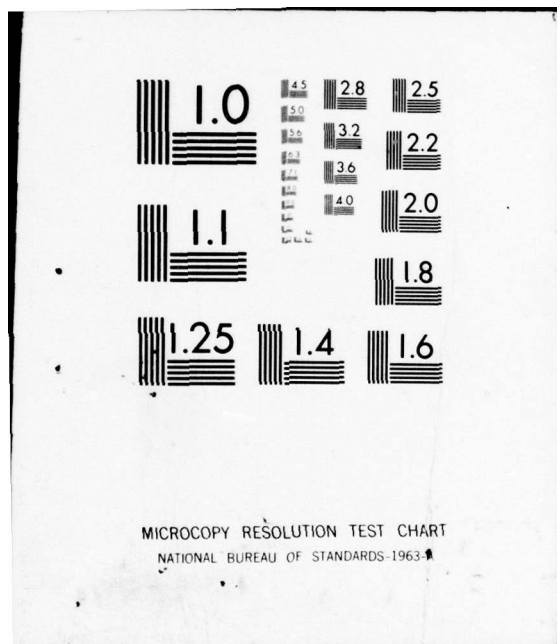
SAI-78-549-WA

NL

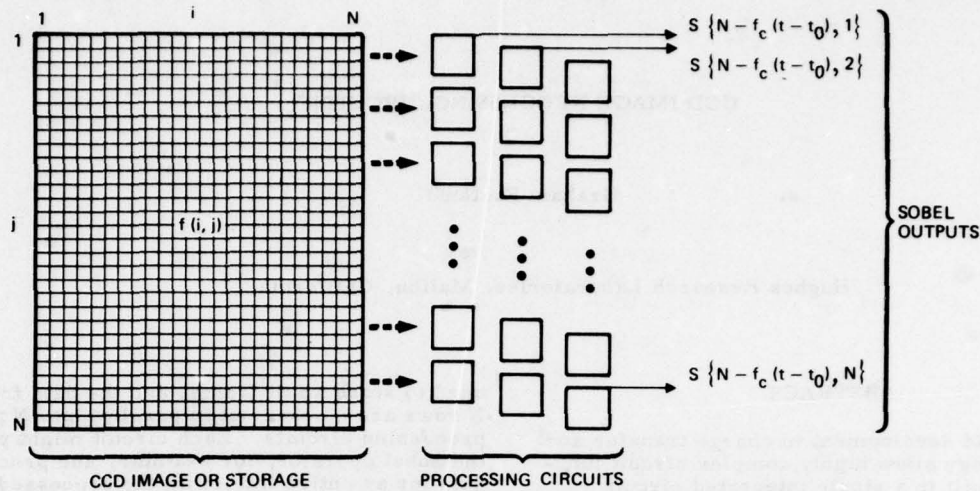
2 OF 2  
AD  
A052900



END  
DATE  
FILMED  
5-78  
DOC







$f_c$  = CCD CLOCK RATE

$t_0$  = START OF FRAME TRANSFER

Fig. 1. Concept of Parallel Pre-Processing Configuration

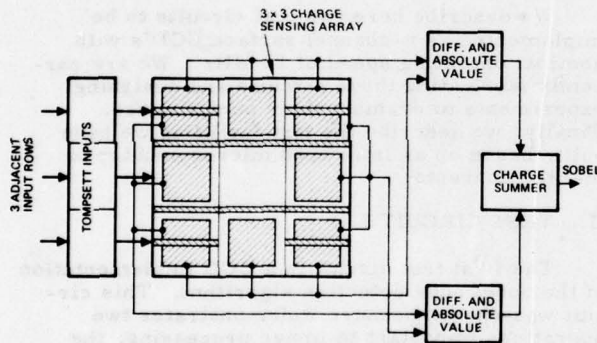


Fig. 2. Schematic of CCD Sobel Circuit

intensities, are fed into the device using Tompsett potential equilibration inputs for linear operation. The top and bottom lines of charge are then divided into two parallel channels using a central implanted channel stop as illustrated, and floating gate electrodes are used to nondestructively sense the charge in each channel. With the electrode configuration shown, the voltage appearing on the top interconnection, for example, is

$$V_1 = k C_{ox} \{f(i-1, j-1) + 2f(i, j-1) + f(i+1, j-1)\}$$

where  $C_{ox}$  is the oxide capacitance and  $k$  is a constant relating the charge generated by the input circuit to the pixel intensity. The weightings, (1, 2, 1) are obtained directly by making the central electrodes twice the area of those on the corners. The voltages appearing on the other three interconnects are equivalent to the other expressions shown in equation (1).

To calculate the full Sobel,  $\{S_{i,j}\}$ , pairs of these outputs are then subtracted and the absolute

value of these operations taken prior to summation. In the direct implementation conventional MOS differential amplifiers can be used to perform this first operation, and the outputs from these fed to absolute value circuits.

Two absolute value circuits are included on Test Circuit I and output will be available from both. Figure 3 depicts the circuit schematic and potential diagram of a single channel CCD absolute circuit. The circuit uses a fill and spill input system to generate a charge,  $Q$ , proportional to the magnitude of the voltage difference on gates Sig and B2, i.e.,  $Q = C_{ox} (V_{sig} - V_{B2})$ . In this way the B2 electrode is used as reference. For a negative input signal the potential profile at the silicon surface is as shown in the upper figure. When the diffusion  $\phi_{INA}$  is pulsed, charge flows along the surface and fills the potential well shown. When  $\phi_{INA}$  drops, the excess charge flows across the potential barrier formed under the signal electrode back to the input diffusion. Then as the transfer gates,  $\phi_{OUTA}$ , are clocked, the charge represented by the shaded area is clocked out. For a positive input signal, the potential profile is shown in the lower figure. After the spill and fill operation is completed by again pulsing the input diffusion, charge collects in the well as shown, and the charge indicated by the shaded area is clocked out. If the total gate area of FZ and SIG is designed to be equal to that of B2 and FZ, equal amount of charge will be transferred for positive and negative signals of the same magnitude. Thus, an absolute value function in the charge domain is obtained. This implementation has a number of advantages which will materially affect the performance and accuracy of the circuit. For example, it always provides a 'fat zero' bias charge packet (indicated by the cross-hatched area) to decrease the transfer inefficiency caused by the surface states in the

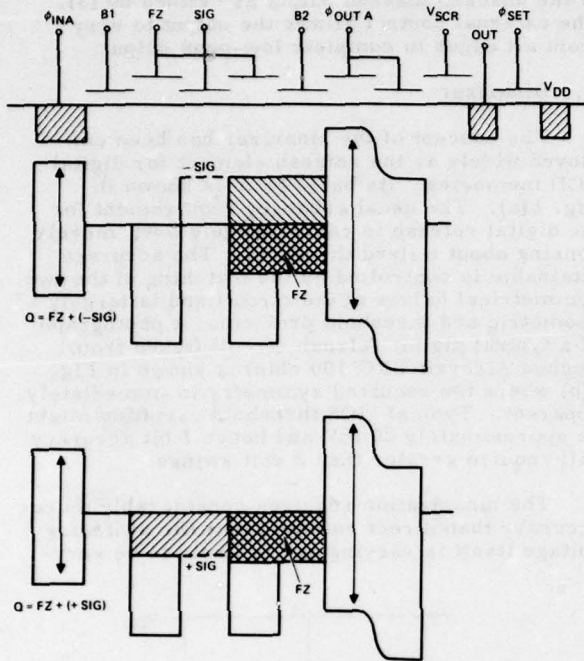


Fig. 3. Absolute Value Circuit No. 1

channel. The level of this 'fat zero' is controlled by the dc bias applied to FZ.

A preliminary experiment was performed on a simple input to demonstrate the functional concept described above. This circuit was not designed for performing absolute value functions. Hence, its input gates are not structured for this particular application. However, it illustrates the validity of the concept. Figure 4 is a scope photograph of both the input and output waveforms. It can be seen that the bottom half of the input waveform is inverted in the output. The output waveform is not symmetrical about the zero level due to the asymmetry of the input gate arrangement.

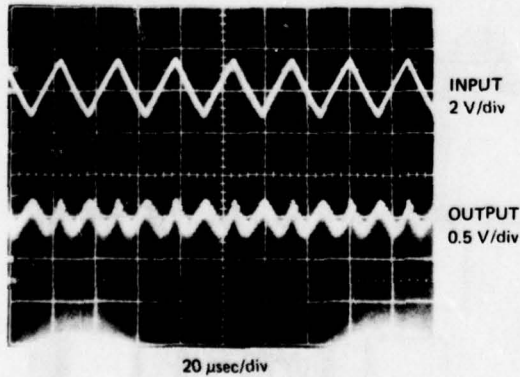


Fig. 4. Input and Output Waveforms of a CCD Absolute Value Circuit

The other absolute value circuit to be included in Test-Chip I uses two parallel CCD channels which act as rectifiers to provide a single difference output. The basic details of this concept were described in the USC Semi-Annual Technical Report for Sept. 30, 1976.

We are currently in the process of fabricating these devices on a Hughes Aircraft Company IR&D chip and we anticipate circuits will be available in April 1977.

### III. TEST CIRCUIT II

The detailed design and layout for a second test circuit is currently in progress. The circuit is designed to operate on a 3 by 3 array of pixels and perform the five operations defined in equations 1 through 5.

$$\text{Low Pass Filter } f_m(i, j) = \frac{1}{9} \sum_{i-1}^{i+1} \sum_{j-1}^{j+1} f(i, j) \quad (2)$$

$$\begin{aligned} \text{Unsharp Masking } S(i, j) &= (1-\alpha) S(i, j) \\ &+ \alpha f_m(i, j) \end{aligned} \quad (3)$$

$$\begin{aligned} \text{Adaptive Binarizer } f_b(i, j) &= \begin{cases} 1 & f_m(i, j) \leq f(i, j) \\ 0 & f_m(i, j) > f(i, j) \end{cases} \end{aligned} \quad (4)$$

$$\begin{aligned} \text{Adaptive Stretching } f_a(i, j) &= \begin{cases} 2 \text{ Min } \{f(i, j), r/2\} & f_m(i, j) \leq r/2 \\ 2 \text{ Max } \{f(i, j) - r/2, 0\} & f_m(i, j) > r/2 \end{cases} \end{aligned} \quad (5)$$

The circuit philosophy is to provide each of the five output functions independently and make the interconnection either with wire bonds on the chip surface or external coax. In this way parallel techniques will be investigated and each function can be isolated and tested separately. For example, two Sobel circuits will be built (one using a HAC proprietary arithmetic technique for charge sensing and calculation), and a number of novel absolute value circuits developed. This will allow us in the initial testing phase to evaluate six different circuit arrangements for edge detection and determine the performance and accuracy of each approach. Then, in the final, image processing, we will select the optimum.

The detailed design and simulation of each of these devices has now been completed. A brief description of each circuit element is given below.

#### A. Edge Detection

The edge detection technique is again based on the Sobel operator, and two circuit concepts are being developed, based on the two dimensional CCD matrix shown in Fig. 2. The design of the



differential amplifiers have been computer simulated up to 20 MHz, and it is estimated that an accuracy equivalent to 7 bits will be achieved with a gain of 0.8. The balance achieved between the two input devices is crucial to accurate operation and in the device now being drawn, particular emphasis is given to this issue.

### B. Low-Pass Filter and Center Element

The low-pass filter uses three floating gate electrodes to sense and sum the charge magnitudes in nine adjacent cells. The output represents

$$\sum_{i=1}^{i+1} \sum_{j=1}^{j+1} f(i, j)$$

and hence is nine times the mean. This has been done (rather than make each floating gate a ninth of the full cell size) to increase the sensitivity. It does, however, require a CCD shift register with nine times the width to sense the center pixel to achieve balanced signals.

### C. Unsharp Masking Circuit

The concept of the unsharp masking circuit is shown in Fig. 5. It is based on the analog multiplier. Externally adjustable inputs (controllable by external power supplies) are fed to transistors T1 and T2 which control the gain of the two input devices T3 and T4. Since these are drawing current from a common source  $V_{DD}$  the voltage of node, N, varies as  $(1 - \alpha) f_s(j, k) + \alpha f_n(j, k)$ . The output from the source follower is thus equivalent

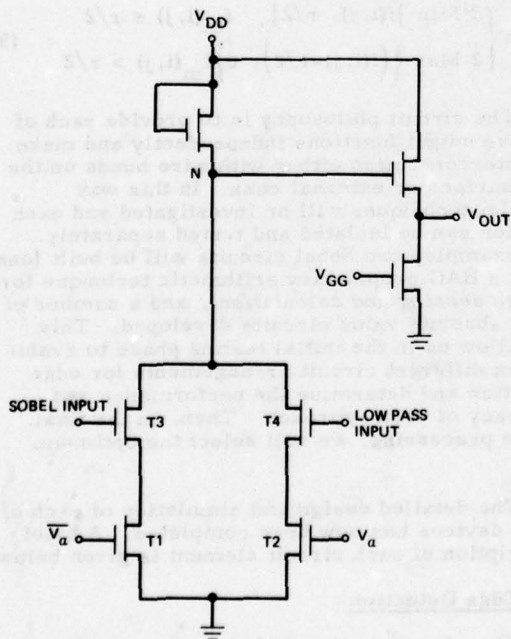


Fig. 5. Analog Multiplier Used for Unsharp Masking

to the unsharp masked output as defined by (3). The external control allows the output to vary from all edges to complete low-pass output.

### D. Binarizer

The concept of the binarizer has been employed widely as the refresh element for digital CCD memories. Its basic form is shown in Fig. 6(a). The usual accuracy requirement for the digital refresh is comparatively low: merely sensing about a fixed threshold. The accuracy attainable is controlled by the matching of the two symmetrical halves of the circuit and is largely a geometric and threshold problem. A photograph of a typical digital refresh circuit (taken from Hughes Aircraft CRC 100 chip) is shown in Fig. 6(b) where the required symmetry is immediately apparent. Typical MOS threshold variation might be approximately 20 mV and hence 7 bit accuracy will require greater than 2 volt swings.

The binarization requires considerably more accuracy than direct refresh since the switching voltage itself is varying and is likely to be very

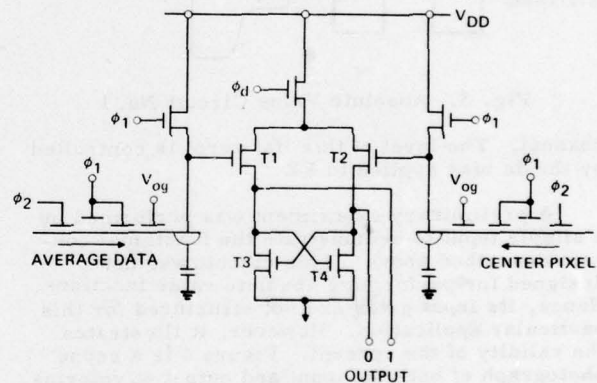


Fig. 6(a). Schematic of Binarizer

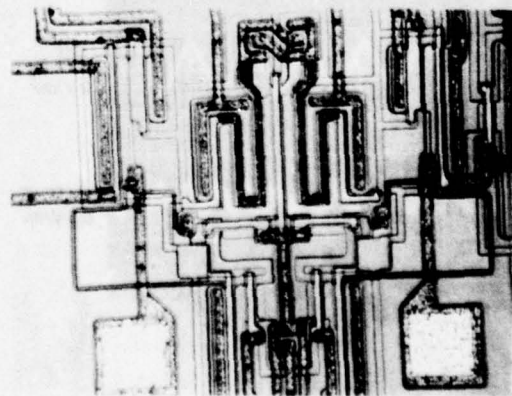


Fig. 6(b). Photo-micrograph of Binarizer



close to the input signal (one being the center pixel, the threshold being the average of its nine neighbors). We are therefore currently considering using a preamplification stage prior to the cross-coupled latch shown. An amplification of say 5 would be sufficient to achieve the necessary accuracy and provide correct latching. This problem is currently being analyzed.

#### Adaptive Stretching

The adaptive stretching function is implemented by having an input signal equivalent to  $f(i, j)$  ac coupled to a MOS transistor which is driven by an external voltage  $V_t$ . (This input can also be derived from the mean  $f_m$  by internal bonding on the chip.) The gain of this circuit is 2 and the output will be linear until the transistor  $T_1$  limits at  $f(i, j)_{\max}$  (i.e., input magnitude  $f(i, j)_{\max}/2$ ). The complement of this output is also available which provides a thresholded output (up to  $f(i, j)_{\max}$ ) and then a linear gain of 2. These two outputs provide the transfer function shown on page 160 of the September 1976 Semi-Annual, isolating the high brightness and shadow regions and can be externally varied by controlling the threshold voltage  $V_t$  and the gain, via the source follower input  $V_g$ .

#### IV. TEST FACILITIES

During the past six months we have spent a considerable time developing the test facilities necessary to demonstrate the performance of our CCD circuits on the USC data base. The concept of the system is shown in Fig. 7. It is based on the IMSAI 8080 microprocessor and interfaces with the USC PDP-10 via a standard 30 byte telephone line. Image data, stored on magnetic tape at the Image Processing Institute, is read by the PDP-10 and transmitted to Hughes Research Laboratories via the existing telephone tie lines, and stored in the digital memory of the microprocessor. The data can then be displayed on the TV monitor shown, and if required stored on a commercial tape recorder cassette for later reference. An eight bit digital to analog converter is then used to access the data in the memory and interface with CCD circuits. The processed data from the circuits is then returned to the memory via an analog to digital converter as shown.

The circuits themselves are bonded in a 40 pin dual in-line package and mounted in a coaxial breakout box, through which the clocking pulses, biases and resets are applied. At the present time all the components shown in Fig. 7 have

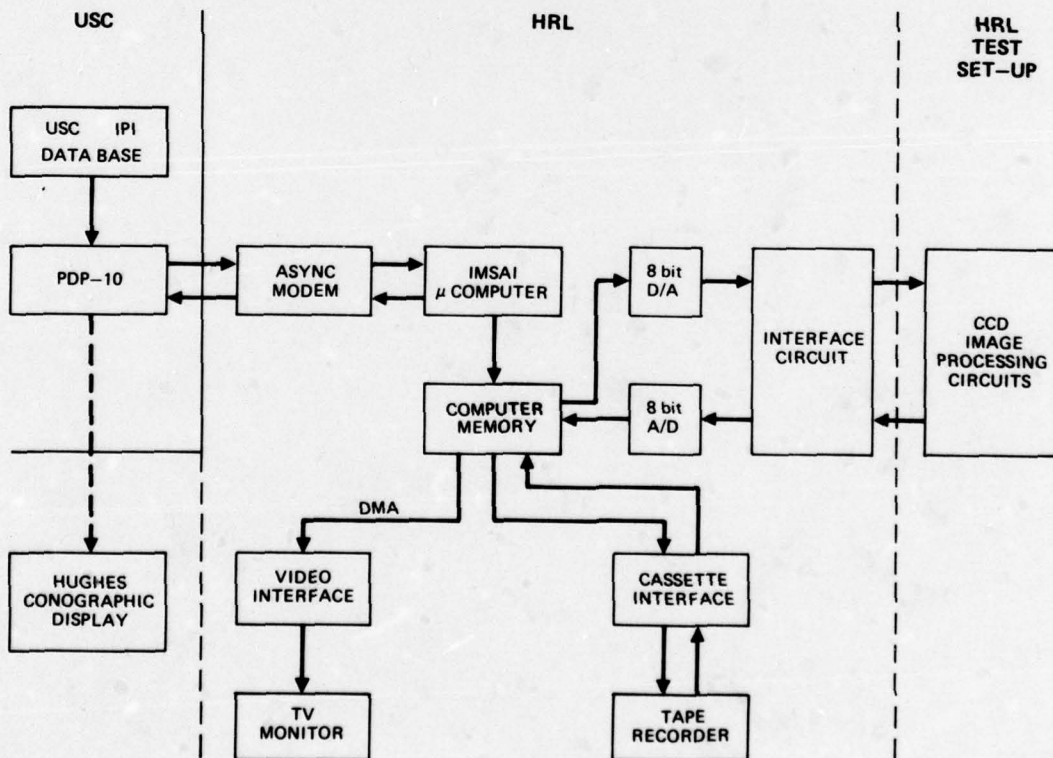
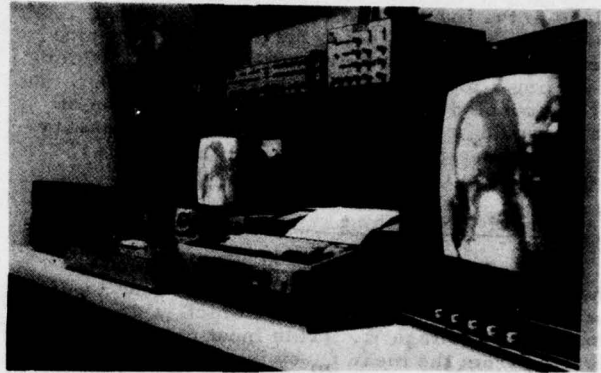


Fig. 7. Schematic of Test Set-up

been built and interfaced to form the full system. A photograph of part of the system is shown in Fig. 8. We have also developed the necessary software to interface the PDP-10 with our system, and successfully accessed images from the USC system for both storage and display.

## ACKNOWLEDGMENT

This report represents the work undertaken in the second phase of the subcontract to Hughes Research Laboratories from USC Image Processing Institute. Participants in the work described are R. Harp, C. L. Jiang, W. Jensen, N. Maeding, P. Nygard, and P. Prince.



**Fig. 8. Photograph of Test Set-up**

## IMAGE UNDERSTANDING RESEARCH AT CMU: A Progress Report

Raj Reddy  
Department of Computer Science  
Carnegie-Mellon University  
Pittsburgh, Pa. 15213  
March 27, 1977

### INTRODUCTION

The primary objective of our research effort is to develop techniques and systems which would lead to successful demonstration of image understanding concepts over a wide variety of tasks, using all the available sources of knowledge. This requires the determination of the type and nature of knowledge that might be applicable in a given task situation. The representation, use, and evaluation of such knowledge must be made within a total system's context. The research program at CMU is an attempt at parallel development of various components, incrementally leading to increasingly complex image understanding systems.

### SYSTEMS AND TASKS

The image understanding research at CMU uses DEC System 10/80, C.mmp (a 16 processor multi-mini computer system), and a dedicated MIPS (Multi-sensor Image Processing System) computer. A complete description of MIPS, including the rationale for various design choices is given in McKeown and Reddy (1977).

Our present plans are to attempt to interpret uncontrived arbitrary images representing different views of the downtown Pittsburgh area (a 3-D world), and aerial and satellite views of the Washington, D.C. area (a 2-D world). The world models for these tasks are expected to be generated incrementally over the next few years.

### KNOWLEDGE REPRESENTATION AND SEARCH

The paper by Rubin and Reddy in this workshop presents our current views about representation of knowledge. The PPE graph structure representation of knowledge tends to be expensive in terms of space required, but is essential if we wish to use the faster beam-search techniques for image interpretation. We expect to embed this particular knowledge representation and search as the principal component into a total system which will involve planning (solution in simpler, coarser, or abstract spaces), iterative dynamic refinement of knowledge representation, and goal-directed interpretation strategies.

At present we are developing the following knowledge sources for the downtown Pittsburgh task: a 3-D model of the downtown Pittsburgh area, knowledge about building structures and textures, knowledge about local refinements given coarse recognition (e.g., detecting cars in roads and trees and bushes next to roads), knowledge about shadows occlusions and highlights, and so on. Given our basic approach of iterative refinement of knowledge, we will start with simple versions of these knowledge sources, and refine them as we observe their limitations when applied to different scenes.

### CHANGE DETECTION

We plan to continue experiments in symbolic

registration and change detection (see the paper by Price and Reddy in this workshop). As changes due to perspective and scale become more and more dominant, it becomes desirable to view the problem of registration as one of search involving constraint satisfaction based on spacial relationships. We think the model presented in Rubin and Reddy (in this workshop) would also be useful in this case. The paper by Kober et al. (in this workshop) from CDC indicates the progress to date on the cooperative image registration research.

### IMAGE DATABASE

If we are to have adequate performance and error analysis tools and tools for knowledge source generation, it is desirable to manually (or interactively) generate symbolic descriptions of the images to be analyzed. This and other considerations have led us to begin to develop a unified symbolic and signal image database system. The structure of this database is described in McKeown and Reddy (1977). The database has several hundred images but only a few have symbolic descriptions so far.

### ARCHITECTURES FOR IMAGE PROCESSING

It is estimated that we will require processing power of the order of 1 to 10 billion instructions per second in an all digital image processing system with rapid response times. We are attempting to develop (in cooperation with CDC) new problem-oriented high speed digital processor architectures for image processing. Given that C.mmp and MIPS are closely coupled multiprocessing systems, we are exploring issues of algorithm decomposition and parallel-pipeline system structures for image processing. Another aspect under study is the development of a special instruction set for image processing using the writable microstore available with the PDP-11 processors on C.mmp and MIPS.

### KNOWLEDGE ACQUISITION

Given the paucity of ideas about type and nature of knowledge used in visual perception, we are continuing our protocol analysis studies in human visual perception. Studies in progress include picture puzzles (Akin and Reddy, 1977), perception as a function of distance, perception in the presence of contradiction, and peep-hole perception studies.

### CONCLUSION

The research program at CMU has many facets, but we expect that progress will be slow given the inherent complexity of the problem and limited resources (present level of effort: about one person per topic above). At present there is very little low-level vision research except for the components generated by Ohlander and Price as part of their theses. We expect to primarily concentrate our efforts on total system design, knowledge acquisition and representation, and specialized architectures for image understanding research.

### REFERENCES

- D. M. McKeown, Jr., D. R. Reddy (1977). "A Hierarchical Symbolic Representation for an Image Database," *Proceeding of IEEE Workshop on Picture Data Description and Management*, April, 1977.
- O. Akin, D. R. Reddy (1977). "Knowledge Acquisition for Image Understanding," to appear in *Journal of Computer Graphics and Image Processing*, 1977.



## 1976-77. PROGRAM REVIEW

R. Larson

HONEYWELL INC.  
Systems and Research  
Minneapolis, Minnesota 55413

The Honeywell contract began in May 1976 and is part of a long range plan to develop a context dependent image screening device. Potential applications for such a device include missile guidance, automatic aerial reconnaissance, RPV data compression, intelligence gathering, FLIR operator cueing and many more. Each application has different requirements for the kind of result the screener must provide and the size and kind of hardware that can be used. However, the basic algorithm and hardware technology for any one mission should be useful to the other missions.

Our current contract has two main parts:

- Autothreshold hardware modification to the Autoscreener.
- Application of syntactic pattern recognition methods.

## AUTOTHRESHOLD HARDWARE MODIFICATION

The autothreshold work is funded by the Air Force Avionics Laboratory and is to make the Autoscreener self adaptive to background and contrast changes. The adapting is done by estimating the background intensity at each pixel position while the picture is being scanned. To make the estimate, the device decides whether the pixel is like the background estimate for that position. If not, then the background estimate is left unchanged. If it is like the background estimate, then the background estimate is updated using the new value. The new background estimate is then compared with pixels in the next scan line. If the actual image intensity is much different than the background estimate, it indicates a possible object of interest. Edge detection is done using the 3x3 Sobel gradient operator. Large values of the gradient give another indication of a possible object of interest. When both the intensity and the gradient indicate an object of interest then that portion of the image is extracted and analyzed further by autoscreener.

This autothreshold algorithm is currently being implemented in hardware to provide a real-time interface between a FLIR sensor and the Autoscreener. Both the background estimate and the edge detection are done in hybrid, discrete analog fashion (continuous amplitude, discrete in space) using CCD scan line storage devices. The current implementation uses discrete hardwired components and occupies five 4½ inch by 6½ inch boards.

## APPLICATION OF SYNTACTIC METHODS

This part of the contract is being funded by DARPA through AFAL and is concerned with developing and applying algorithms for real time image screening. The initial syntactic pattern recognition effort has three aspects:

- Define a tractable problem, select approaches to its solution and obtain a suitable data base for the work.
- Develop Autoscreener relevant DARPA research at Purdue University.
- Evaluate and apply Honeywell IR&D and FLIR research to the Autoscreener.

## Problem Definition

In consultation with Professor T.S. Huang of Purdue, we decided to direct the first year toward recognizing airborne FLIR images of isolated tactical targets in a rural environment. FLIR imagery was chosen (rather than photographs, TV and downward looking IR) as being the most likely tactical sensor. The class of isolated tactical targets was deemed sufficiently complex and large for a first attempt at applying syntactic and contextual methods to tactical quality imagery. We chose to further limit our attention to those kinds of isolated tactical targets for which we could obtain suitable data.

Data Base Selection--The Krebs data base\* was chosen as our primary data source. The Krebs data contains a large variety of tactical targets of both military and non-military types. In addition to ground vehicles, there is imagery of factories, docks, bridges (short and long), power lines, houses and even one example of a helicopter flying across the field of view. After much discussion, we decided to work only with military ground vehicles as the primary target class and include as additional classes of interest any objects or background areas that appeared in frames containing primary targets and which could either aid classification by context or which might be confused with the primary targets. The object types selected include: Tank, Truck, APC, Car, House, Road, Vegetation, Shadow.

Solution Approach--Next, we considered possible methods of recognizing these target types. In general, an imagery recognition system has four levels of activity:

1. Object Detection
2. Target Detection
3. Target/Background Separation (Boundary Location)
4. Target Recognition

Each level of activity has its own goals and the degree to which it meets these goals affects the performance of the following levels. The functions in each level are also affected by the size of the target image. Because of the variety of functions and performance requirements and the fact that parts of the problem have been studied and/or solved by other efforts, the scope of the approach was limited to points 3 and 4 - Target/Background Separation and Target Recognition - and we decided to focus on medium resolution imagery. (Medium resolution is defined as vehicle image area of 50 to 600 pels.)

#### Autoscreener Relevant Research at Purdue

Purdue University has been looking at the FLIR tactical target recognition problem with the intent to combine various simple measurements into a "smart detector" using texture, shape, and context information. Several basic simple measurement techniques being developed at Purdue have been applied to the FLIR imagery to determine what types of processing are useful in a higher level system. These include texture segmentation, statistical contour following, and feature-plane clustering. The overall detector

design is now being developed which will combine the useful measures intelligently. Details of this work are reported separately by Purdue.

#### Applications of Honeywell Technology

The group at Purdue had prior experience with using high-level primitives (image segmentation, shape analysis, texture transformations, etc.) in image analysis. Honeywell therefore chose to transform the imagery directly into a low-level symbolic representation and perform syntactic analysis on the resulting symbol array. The method chosen was a method for adaptive cluster seeking that had been conceived in signal processing research. The method was restated for use on imagery and applied to the digitized FLIR data. The results of this work are described in the workshop paper "Symbolic Analysis of Images using Prototype Similarity".

\* The Kreb's data set was collected under RADC funding using a Honeywell 18 detector serial scan FLIR mounted in a Navy P2-V aircraft flying out of the Naval Air Test Center at Patuxent River, Maryland. The flights were made during the spring of 1974 at altitudes from 2500 feet to 3500 feet. The original purpose for the data was to evaluate human factors issues in operator recognition of FLIR targets.



## ALGORITHMS AND HARDWARE TECHNOLOGY FOR IMAGE RECOGNITION

Project status report - March, 1977

Computer Science Center  
University of Maryland  
College Park, MD 20742

## ABSTRACT

This report summarizes the current status of the research being conducted under Contract DAAG53-76C-0138 (DARPA order 3206), as well as plans for work to be done on this project in the near future. This project was initiated on May 1, 1976. It is being carried out by the Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD; Profs. Azriel Rosenfeld and David L. Milgram are principal investigators. It is devoted to the development and selection of algorithms for automatic target cueing on Forward-Looking InfraRed (FLIR) imagery, and to the hardware implementation of one or two such algorithms. The hardware aspects are being investigated by the Westinghouse Defense and Electronic Systems Center, Systems Development Division, Baltimore, MD; program director of this subcontract is Dr. Glenn E. Tisdale. The project is being monitored by Messrs. John Dehne and George Jones of the U. S. Army Night Vision Laboratory, Ft. Belvoir, VA.

## 1. Introduction

The project reviewed in this status report has two principal goals:

- a) Selection of state-of-the-art algorithms for automatic target cueing, and implementation of one or two selected algorithms in hardware to demonstrate the feasibility of incorporating such algorithms in a reconnaissance sensor.
- b) Exploration of new approaches to image understanding, with emphasis on techniques applicable to target cueing and similar applications, as well as on image modeling for performance prediction.

The project consists of three phases all of which involve collaboration between the University and its subcontractor, the Systems Development Division of Westinghouse. The three phases and their breakdown into tasks are displayed in the following table:

Phase	Task
I	(Task and technology review)
	1) Data base acquisition
	2) Review of tri-service operational needs and resulting system design constraints.
	3) Hardware/algorithm interface
II	(Algorithm development and testing)
	4) Algorithm development
	5) Algorithm selection and test
	6) Target and background modeling
III	(Hardware design, fabrication, and testing)
2.	Data base acquisition
	Three data bases were acquired and preprocessed (smoothed, windowed):
	a) NVL data base: A set of 145 FLIR scenes containing targets (tanks, trucks, APC's) against a sparsely wooded or barren background.
	b) Alabama data base: A set of 54 FLIR scenes containing targets (tanks, APC's, buses, jeeps, and personnel) against a somewhat less noisy background.
	c) Sequential data base: A sequence of 10 FLIR images, similar to those in the first data base, taken 1/15 second apart.

Detailed descriptions of the first data base can be found in [1], and of the second and third data bases in [2].

## 3. Image modelling

An approach to modelling FLIR imagery has been developed, based on the simplifying assumption that targets appear as homogeneous hot regions within a homogeneous cooler surround. This model describes the joint probability density



of gray level and edge strength in such images, for various edge-detecting operators [1, 2]. In brief, the model predicts that for low edge values (corresponding to points in the interiors of objects and background), there should be two relatively well separated probability peaks, of different sizes, representing the gray levels of object and background interiors, respectively. For higher edge values, corresponding to points on object/background borders, these peaks should move together and become a single peak representing the border range of gray levels.

The model just described can be used as a guide to segmenting FLIR images by thresholding. At low edge values, it should be easy to pick a threshold at a gray level in the valley between the two probability peaks, since these are relatively well separated. At high edge values, the peak gray level value itself, or perhaps the mean gray level, should be a good threshold, since this represents the "center" of the edges. For intermediate edge values, one can compromise between these two thresholds in various ways. A comparative study of threshold selection schemes based on this approach has been conducted [3], and has shown them to be superior to conventional threshold selection methods.

#### 4. Object extraction based on threshold selection

As a preliminary to using thresholding to extract objects from an image, it is important to smooth the image, so that the extracted objects will not be too noisy. The use of both mean and median filtering for this purpose was investigated [1-2]. It was found that median filtering using a 3x3 neighborhood of each point produced the best results. An adaptive technique, which identifies neighborhoods that are noisy and edge-free, was shown in [4] to smooth noisy regions in images without degrading edges. The technique was also used to produce a weighting function to suppress spurious responses of an edge detector operating in a noisy environment.

Threshold selection based on the (gray level, edge strength) probability density was investigated using a number of different edge detectors [1-2]. It was found that a "coarse gradient" detector, based on differences of averages taken over pairs of adjacent 4x4 neighborhoods, gave the best results, since this detector combines smoothing with edge detection.

Even when applied to smoothed images, thresholding methods will sometimes extract noise regions, as well as objects. Postprocessing techniques, based on

shrinking and reexpanding of the above-threshold areas, can be used to eliminate many of these noise regions. Several variations on this approach were studied in [1-2].

The regions surviving the postprocessing step must now be classified into target and nontarget classes. To this end, the connected components of the surviving points are extracted, and a set of size and shape features is measured for each component. A discussion of feature selection and the classification procedure will be presented in Section 6.

#### 5. Object extraction based on edge/border coincidence

The thresholding approach to object extraction described above has the disadvantage that a single threshold will usually not be satisfactory for an entire scene. If the image is divided into windows, two difficulties arise if a threshold is assigned to each window. First, it is still possible that objects at different intensities will be in the same windows and second, an object may now overlap several windows. In the former case, we will miss one or more objects (assuming the threshold selection algorithm will choose an appropriate threshold for the multi-object window). In the latter, different thresholds may have been chosen in adjacent windows to extract regions of the same object. This inconsistency is likely to affect what the thresholded objects look like. A further difficulty attends the interpretation of the thresholded image since it becomes difficult to differentiate object regions from noise regions.

The approach which has been developed views the extraction of objects as a classification process into two classes: object regions and noise regions. Regions to be classified are extracted by first thresholding the (smoothed) image and then segmenting the thresholded image into connected components. Each connected component is considered to be a candidate for classification. Three heuristics are used: a size heuristic, a contrast heuristic and a "well-definedness" heuristic. If object size range is known a priori, then noise regions outside the object size range can be rejected. The contrast heuristic states that objects contrast with their surrounds. This may be quantified by measuring the average gray level difference between the interior of a connected component and its boundary. Finally, the well-definedness heuristic states that objects are viewed as being distinct from their surround by the presence of an edge at the boundary. This is computed first, by extracting an "edge map" from the scene, consisting of the

result of thinning the output of an edge detector; second, by measuring for each extracted region the percentage of its border which coincides with the edge map.

The combination of the contrast measure with the edge/border coincidence serves both as a discriminant function for object regions and as a figure of merit for ranking the classified object regions. This approach does not require the user to preselect a particular threshold or set of thresholds. However, the speed of the algorithm is linear in the number of thresholds investigated. Moreover, the false alarm rate is related to the gray level probability of the chosen thresholds. This implies that care in selecting thresholds will generally be worthwhile. An implementation of this method has provided good segmentations of FLIR windows.

#### 6. Target classification

Regions classified as objects by the methods of Section 5, may be further classified as to target type. A hierarchical decision structure has been implemented based on size, shape and contrast features. Object regions which survive the prescreening are divided into two groups based on size. The group of smaller regions is classified into target and noise classes based on compact shape and contrast. No attempt is made to identify the particular target types since these objects generally correspond to vehicles at long range with no identifiable characteristics. The group of larger regions is classified into tank, APC, truck and noise classes based on shape (compactness, symmetry, aspect) and contrast.

Selection of the set of features actually used at each node of the decision tree is restricted to those "logically allowable" at the given node. For example, while the brightness of a region is allowed to distinguish objects from noise, it is not used to determine vehicle type. The point of this restriction is to reduce the dependence of the final classifier on the pre-classified data, increasing both the robustness and the intelligibility of the classification. After this logical preselection is made, the effectiveness of features to be assigned to a node can be evaluated by standard statistical techniques (analysis of covariance, multiple discriminant analysis). The purpose is to increase the stability of the classifier without decreasing its accuracy. Experiments described in [2] exhibit good self-classification; however, we have not obtained good results when extending the classifier to a test set.

#### 7. Hardware design

The Westinghouse Systems Development Division as a subcontractor to the Univer-

sity of Maryland has concentrated on the hardware implementation and fabrication of image algorithms for the focal plane [1, 2]. Algorithms whose hardware implementation has been designed include: median filtering, edge detection using differences of averages, edge thinning by non-maximum suppression, threshold selection based on a (gradient, gray level) histogram, noise cleaning by shrinking and expanding and additional support logic such as serpentine delay lines and A/D converters. The attempt throughout is to design and build algorithms in analog CCD hardware within overall system constraints on data flow, storage requirements, chip-size, yield factors and cost.

#### References

- [1] Algorithms and Hardware Technology for Image Recognition, First Quarterly Report, Computer Science Center, Univ. of Maryland, College Pk., MD, July 1976.
- [2] Algorithms and Hardware Technology for Image Recognition, First Semi-Annual Report, Computer Science Center, Univ. of Maryland, College Pk., MD, October 1976.
- [3] D. P. Panda, Segmentation of FLIR Images by Pixel Classification, Computer Science Center, Univ. of Maryland, College Pk., MD, Tech. Rep. 508, Feb. 1977.
- [4] D. P. Panda, A Method of Adaptive Smoothing and Edge Enhancement, Computer Science Center, Univ. of Maryland, College Pk., MD, Tech. Rep. 504, Feb. 1977.
- [5] D. L. Milgram, Region Extraction Using Convergent Evidence, Proceedings of the ARPA SemiAnnual Workshop on Image Understanding, Minneapolis, MN, April 1977.



## IMAGE UNDERSTANDING AND INFORMATION EXTRACTION

K.S. Fu and T.S. Huang  
Purdue University  
W. Lafayette, Indiana 47907

This is a progress report of our research in Image Understanding and Information Extraction during the last six months. The objective of this research is to achieve better understanding of image structure and to improve the capability of image data processing systems to extract information from imagery and to convey that information in a useful form. The results of this research are expected to provide the basis for technology development relative to military applications of machine extraction of information from aircraft and satellite imagery.

Our research projects can be categorized into six rather heavily overlapping areas. Image Segmentation, Image Attributes, Image Structure, Image Recognition Techniques, Preprocessing and Applications. The relationships and interactions among these categories is suggested by Figure 1. After the sensor collects the image data, the preprocessor may either compress it for storage or transmission or it may attempt to put the data into a form more suitable for analysis. Image segmentation may simply involve locating objects in the image or, for complex scenes, determination of characteristically different regions may be required. Each of the objects or regions is categorized by the classifier which may use either classical decision-theoretic methods or some of the more recently developed syntactic methods. In linguistic terminology, the regions (objects) are primitives, and the classifier finds attributes for these primitives. Finally, the structural analyzer attempts to determine the spatial, spectral, and/or temporal relationships among the classified primitives. In some respects, this is where real "image understanding" is developed.

Our accomplishments during the past six months have been recorded in our progress reports [1,2]. Here we shall summarize the highlights:

**IMAGE SEGMENTATION** - Considerable progress has been seen in segmentation of imagery by clustering methods. Yoo and Huang have pursued this approach throughout the year, and a summary of their work, with application to images containing a tank and two aircraft, is included in [1].

The work of Carlton and Mitchell concerning texture has now evolved from the study of image attributes to the development of techniques for image segmentation using texture and gray level features. Their results, discussed in [1], offer

the potential of a very fast method which might be implemented either digitally, optically, or with CCD devices.

Keng and Fu have studied the problem of image segmentation by a syntactic method. This method involves the following four steps: (1) texture region primitive extraction, (2) boundary primitive extraction, (3) grammatical inference, and (4) syntax analysis. Examples of applying the method to various images are also reported [2].

**IMAGE ATTRIBUTES** - Following our earlier effort on Fourier shape descriptors, Wallace and Wintz have continued to study the use of Fourier descriptors for three-dimensional objects. Yoo and Huang have investigated the sufficient number of Fourier coefficients for a discrimination process. An efficient implementation of Fourier descriptor algorithms is also described [2].

**IMAGE STRUCTURE** - A syntactic approach to shape description has been studied by You and Fu. A 4-tuple curve primitive and an angle primitive are proposed, and their properties studied. Shape grammars based on the proposed primitives are constructed for different shapes [2].

**IMAGE RECOGNITION TECHNIQUES** - A "supervised clustering" method has been shown by Fukunaga and Short to be useful for localizing a problem rather than dealing with a more difficult global problem. Computationally simple yet accurate results are obtained. Potential applications of the approach include linear classifier design and density estimation [1].

Classification using image context has been studied by Swain and Kit. A statistical contextual classifier minimizing Bayes risk is derived. Preliminary results from data simulation have been reported [2].

**PREPROCESSING** - The projection algorithm for image restoration studied by Berger and Huang is being adapted for use on actual satellite data. Practical considerations of the algorithm are reported [1,2]. O'Connor and Huang have investigated the phase unwrapping with applications to stability and picture deblurring. Improvements in one-dimensional phase unwrapping are made and a two-dimensional phase unwrapping algorithm is proposed. Extensive results from the development are reported [2].



APPLICATIONS - Fourier descriptors have been demonstrated to be a useful means of describing the shape of a closed planar figure, and, in particular, Wallace and Wintz have used Fourier descriptors to encode the shapes of aircraft. Results using this approach for aircraft recognition have been reported [1].

Spatial filtering has been used by Mitchell, et. al. to reduce the effects of light cloud cover in satellite imagery. Results from a computer simulation and from LANDSAT data are discussed in [1].

#### REFERENCES:

1. T. S. Huang and K. S. Fu, "Image Understanding and Information Extraction," Final Report, Nov. 1, 1975 - Oct. 31, 1976, DARPA Contract No. F 30602-75-C-0150, Purdue University, W. Lafayette, Indiana.
2. T. S. Huang and K. S. Fu, "Image Understanding and Information Extraction," Quarterly Progress Report, Nov. 1, 1976 - Jan. 31, 1977, Purdue University, W. Lafayette, Indiana.

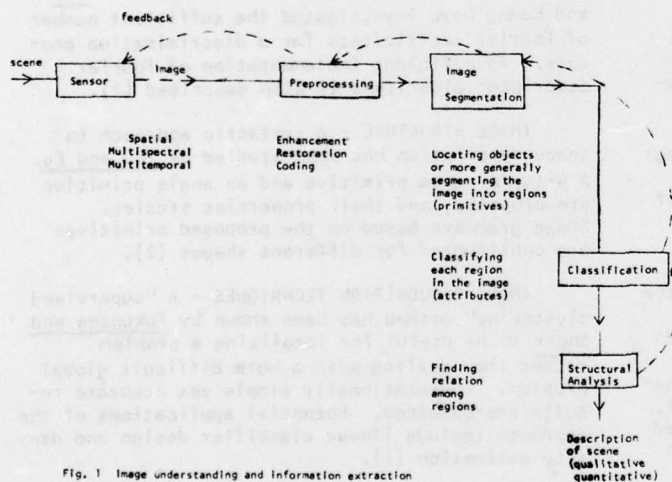


Fig. 1 Image understanding and information extraction

## OVERVIEW OF THE ROCHESTER IMAGE UNDERSTANDING PROJECT

Jerome A. Feldman

Computer Science Department  
The University of Rochester

The Rochester program covers three major sub-topics: using map knowledge for specific aerial imagery tasks, development of a general image understanding framework, and system support for image understanding. These will be briefly discussed in reverse order.

One of our first tasks was the extension of the SAIL programming language which is heavily used by contractors in this program. The largest change was increasing the number of allowable items from 4,000 to 256,000, thus overcoming one major bottleneck. This addition, along with others which increase speed and reliability, was incorporated into the standard SAIL files at Stanford and have been distributed to user sites. Current programming language work is centered around development of a new language for distributed computing tasks, including image processing.

The entire problem of dealing with images in a network environment is of continuing concern. We have made considerable progress towards a flexible, machine-independent distributed image processing system. A summary of this work and references to more detailed descriptions can be found in the technical paper by Feldman and Rashid in this volume. One result of this work should be a domain-independent interface between image displays and image understanding programs.

The development of these general tools is part of a larger effort to construct an image understanding system which will be useful for a wide variety of tasks. This work is proceeding in parallel with our direct attacks on practical image understanding tasks for ARPA and other agencies. The interaction between tool builders and tool users is having the multiplier effect we expected when we started these efforts. An overview of the general vision system is given in the paper by Brown and Lantz in this volume.

We want the system to be a practical aid to accomplishing vision tasks rather than just a methodology in search of a problem. It is therefore initially designed to act as a framework in which to answer specific "queries" about images posed as domain-dependent user code. At the same time we want to develop and incorporate generally

useful facilities for control of vision tasks, representation of knowledge, and automated reasoning. As the latter facilities develop, the user code may be written to leave more and more details to the system. All of this system and generalization work is directed towards the carrying out of specific image understanding missions. Our immediate goal is to use a variety of kinds of knowledge to solve particular problems in aerial imagery. For example, consider the problem of finding and classifying ships in an aerial image of a known port. Details are given in the paper by Brown and Lantz, but the main thrust of the approach is to use prior knowledge about where and how ships may appear to direct search for them. We want to have available through the system map knowledge about the source of the image (e.g., where coastlines are), assertional knowledge about how ships look, where (in relation to other objects) they are found, and procedural knowledge (e.g., how to verify the presence of a given shape). The system is designed to facilitate effective use of this diverse knowledge, and furthermore to provide facilities for the more-or-less automatic performance of common tasks such as selecting the best procedure for a task or reasoning about the relative location of objects.

Textural areas can be thought of as those parts of an image where segmentation based on normal similarity measures fails. Meaningful analysis of textured areas must include discrimination between different textures and detection of parts of the same texture. The similarity of textures which are identical except for a scale change, a rotation, or a different range of intensities must be recognized. Standard texture analysis techniques rely on the calculation of a set of features (like edge probability per unit area, or local neighborhood co-occurrence probability matrices) on training sets of images, taking statistical measures of these features for each training set (mean, standard deviation, entropy, etc.), and partitioning the feature hyper-space so that each partition contains exactly one training set. Unknown texture patches are now measured by the same feature operators to determine their location in feature hyper-space, and are assigned the texture class of the appropriate partition. This technique works well for limited domains, where an accurate

training set can be chosen, and where textures exhibit variation in the local features measured. Rotations and scale changes result in a new texture class assignment.

We approach the texture problem by dividing texture regions into meaningful sub-elements of similar intensity sample points, then using rotation- and scale-invariant shape measures to characterize these regions, and finally determining spatial relationships among our sub-elements. By using a decision-tree program structure, easily discriminated textures are separated quickly, and more complex textural structure is only extracted when necessary. This texture analysis scheme not only classifies texture patches into sets, but also produces a description of similarities and differences among different patches. That information is then available to higher-level semantically driven processes, and is more useful than a binary same/different decision.

Much of our early effort has been devoted to bringing up a system which would enable us to pursue these tasks. Then we set out to gather as much relevant software and example imagery as possible to avoid duplication. This has been quite successful due to the common use of the SAIL language and the great cooperation by other contractors. We are especially grateful to the groups at CMU, Stanford, SRI, and USC for helping us get so much productive work done so soon.



## INTERACTIVE AIDS FOR CARTOGRAPHY AND PHOTO INTERPRETATION:

PROGRESS REPORT APRIL 1976 TO APRIL 1977

H.G. Barrow (Principal Investigator)

Artificial Intelligence Center  
Stanford Research Institute  
Menlo Park, California 94025

## Objectives

The central scientific goal of our research is to investigate and develop ways in which diverse sources of knowledge may be brought to bear on the problem of interpreting images. The research is focused on the specific problems entailed in interpreting aerial photographs for cartographic or intelligence purposes.

A key concept is the use of a generalized digital map to guide the process of image interpretation. This "map" is actually a data base containing generic descriptions of objects and situations, available imagery, and techniques, in addition to topographical and cultural information found in conventional maps.

We recognize that within the limitations of the current state of image understanding it is not possible to replace a skilled photo interpreter. It is possible, however, to greatly facilitate his work by providing a number of collaborative aids that relieve him of his more mundane and tedious chores.

## Progress to Date

## Overview:

Our work has been centered on evolutionary development towards an integrated interactive system. It consists of an interactive display console, a map data base, an image library, general image analysis routines, and task specialist routines. At present the system is not a unified whole, but exists as a collection of programs: we are still working towards their integration. The following scenario illustrates the major capabilities that have been demonstrated to date.

The first task when a new image enters the system is to establish correspondence with the map. This is accomplished automatically, selecting potentially visible landmarks (using navigational data associated with the image) and then locating them in the image using scene analysis techniques. The next step is to confirm the validity of existing knowledge. The system can automatically verify the presence of certain cartographic features, such as roads and waterways, and can also monitor the status of some typical dynamic situations, such as ships berthed in harbor or box cars stored in a classification yard. New features are identified and incorporated into the data base using a number of interactive aids for mensuration and tracing. For example, new roads can be traced, or heights of bridge supports can be measured. The system can now use the data base to answer simple queries, entered by a photo interpreter via keyboard and display cursor, such as, "show me pier 14", "what is this building?" or "how high is that mountain". It also has the capability for responding to a more complex query, such as "how many ships were in Oakland-Harbor yesterday?", by retrieving the relevant image from the library, and then invoking the appropriate task specialist.

At present, the questions that can be asked are limited by the small size of the data base and the available specialist routines. The specialists to date are for carefully chosen tasks that could be performed with existing primitive low level vision capabilities. Moreover, as pointed out earlier, the demonstrated task capabilities do not yet exist as a truly unified system, but as a collection of independent programs that share a common data base. They do, however, show the potential of bringing image understanding and artificial intelligence approaches to bear on problems in cartography and photo interpretation.

### Technical Details:

The first task in the scenario is putting the sensed image into geometric correspondence with reference imagery or a map data base. This is fundamental to virtually every military application of imagery. Our initial approach was a modest improvement on conventional image correlation. Given an image and approximate viewpoint, the system determined potentially visible landmarks, and then retrieved images containing the landmarks from the library. For each landmark, an appropriate area of the reference image was extracted and reprojected to make it appear more similar to the sensed image. The reprojection was accomplished using a camera model, based on calibration data associated with the reference image, and elevation data obtained from the map. Each reprojected image fragment was then correlated in a small predicted area of the sensed image, using Moravec's high speed algorithm. From the pairs of corresponding image and world locations the exact camera parameters for the sensed image were computed by solving an over-constrained set of equations.

Although reprojection prior to matching is an improvement on conventional image correlation, the fundamental limitation of the correlation approach, namely sensitivity to viewing conditions, remains. In particular, it still cannot match images obtained from radically different viewpoints (e.g. low altitude obliques to high altitude verticals), sensors, or seasonal climatic conditions, and it cannot match images against symbolic maps. To overcome these limitations, we developed a new approach, parametric correspondence, for matching images directly to a three dimensional symbolic reference map.

The map contains a compact three-dimensional representation of the shape of major landmarks, such as coastlines, buildings, and roads. An analytic camera model is used to predict the location and appearance of landmarks in the image, generating a projection for an assumed viewpoint. Correspondence is achieved by adjusting the parameters of the camera model until the predicted appearances of the landmarks optimally match a symbolic description extracted from the image. The matching of image and map features is performed rapidly by a new technique, called "chamfer matching", that compares the shapes of two collections of shape fragments, at a cost proportional to linear dimension, rather than area. These two new techniques permit the matching of spatially extensive features on the basis of shape, which reduces the risk of

ambiguous matches and the dependence on viewing conditions inherent in the conventional correlation based approach. The technique is described in more detail in our technical presentation. It has obvious application to navigation and targeting as well as photo interpretation.

Having placed the image into parametric correspondence with the three dimensional map, we are now in a position to predict the image coordinates of any feature in the map, and conversely, to predict the map features corresponding to any point in the image. The former is used, for example, in monitoring to indicate exactly where in the picture to look. The latter is used to facilitate interactive graphical communication between the photo interpreter and the data base. Using the camera model and image calibration, many photo interpretation mensuration tasks may be accomplished simply. Routines exist for determining location, length, height, or straight line distance, for features indicated interactively in the image, as well as velocity for objects (e.g. ships or cars) indicated in two images. The camera model provides a unifying theoretical foundation that subsumes what would otherwise be a collection of ad hoc trigonometric techniques. Combining the map and calibrated image, the system can also determine alternative routes and travel distances along roads between indicated points.

It is important to keep in mind that a map is only an approximation to reality: it may be incomplete, be out of date, suppress details, or contain errors. In order to monitor or to make a detailed interpretation of an image, it is necessary to locate image coordinates of objects more precisely than can be predicted using the map and calibration. We need routines which can take predictions and verify them in the image. As a first step in that direction we developed a guided line tracing routine that accepts a rough approximation to the path of linear features, such as rivers or roads, and extracts a best estimate of the precise path in the image. It operates by applying a specially developed line detector in the vicinity of the approximate path, and then finds a globally optimal path based on the local feature values.

The tracing routine is used in two ways; to verify the presence of known cartographic features, using prediction from the map, and to interactively trace new features for incorporation into the map, using a guideline sketched by the user. The tracing of linear features is currently a tedious manual process that



constitutes a major bottleneck in map production.

Having a map and image in correspondence makes the automation of many monitoring tasks feasible. Keeping track of box cars in a railyard, for example, is a typical tedious photo interpretation task. Knowing the layout of the tracks, makes the task essentially a one-dimensional template matching problem. A routine has been developed which flies statistical operators along a track line to hypothesize possible ends of box cars. These hypotheses are used with knowledge of standard box car lengths and characteristics of empty track to locate the gaps between box cars. The program then reports the number of cars, classified by length.

Estimating highway traffic is a similar problem which could be approached by flying car and truck templates along the path determined by the guided road tracer.

Monitoring the presence of ships in a harbor is particularly easy to automate when the map contains details of berths. Given a question about the status of a particular harbor at a particular time, the appropriate image is retrieved from the data base. The ship monitoring routine then projects berth locations from the map onto the image and uses an edge histogram of that region to determine whether the berth is occupied.

The key to automatic monitoring lies in being able to place the image into correspondence with the map, which then accurately specifies where to look. A relatively simple test may then be used in that limited context. We have implemented three representative demonstrations of this approach and believe that many others are possible. In a production environment, such monitoring could be performed automatically on a continuing basis as new imagery arrived.

The underlying foundation upon which much of the foregoing rests is the map data base. We have implemented a disc-based semantic net data structure which can contain realistic quantities of data represented in a way which permits efficient access. Entities are represented by LISP atoms (cf. English words) and information associated with the entity is stored in a property list format. When information concerning a particular entity is sought, the property list is retrieved from disc and established in core. A "paging" scheme

limits the amount of data in core (to, say, 1000 entities) and writes entities back out to disc, if necessary, least recently used ones first.

We are in process of setting up a map of the San Francisco Bay area, containing major features, coastlines, bridges and highways. The geographic data is indexed (the index structure is part of the database) to enable fast retrieval of information relevant to a particular area. In addition to the three dimensional description of cartographic and cultural features, the map contains a partial taxonomy of world entities, with relevant general semantics, a catalogue of available imagery, and descriptions of data structures used by the system. The latter enables the system to construct automatically new entities of the correct structure for inclusion in the data base.

#### Future Plans

The fundamental problem addressed in our research is putting what we see (i.e. a sensed image) into correspondence with what we know (i.e. the map data base). The geometric type of correspondence used extensively above, is particularly simple because it relies on precise knowledge of the appearance and structural relationships of particular objects and of the viewpoint.

The ultimate goal of automating photo interpretation requires a much more general matching capability. For automatic map updating it must be possible to recognize objects described generically, for example, airports or buildings whose precise form is not previously known. This sets several important requirements. The components of a generic description must be established and a means found for reliably extracting appropriate features from the image; a means of communicating descriptions to the machine more conveniently than programming must be developed; and the capability of using the description in the context of what is already known must be developed.

In terms of these long range objectives, the ad hoc task specialists developed so far are lacking in both generality and robustness. The primitive descriptions they employ are adequate only for straightforward cases, and can easily be fooled by the unexpected. We also need the ability to teach high level descriptions by showing examples, rather than the time-consuming process of programming. Our next objective is, therefore, a system that can be



interactively programmed to find instances of objects whose generic descriptions are taught by example. Such a system would allow a PI, for example to point at a box car of a particular type and ask how many are present in the railyard, or to identify a new class of airplane and ask whether any are visible in earlier coverage (possibly misidentified previously). These capabilities are very hard to automate because of the difficulty of reliably extracting features, so we intend to begin with simpler cases, such as finding airports or buildings. At the same time we intend to experiment with identifying objects from more complex descriptions, relying on the PI to interactively indicate features. A recognition aid developed along the lines of MYCIN would be a significant improvement over conventional state-of-the-art systems which do little more than facilitate mensuration. This work will also help to identify needed feature extraction capabilities as candidates for development.

# THE USC IMAGE UNDERSTANDING PROJECT\*

1 October 1976 to 31 March 1977

Harry C. Andrews

Image Processing Institute  
University of Southern California  
Los Angeles, California 90007

## 1. Research Overview

This document represents the third semi-annual report funded under the current ARPA Image Understanding contract and, as such, presents a certain amount of momentum and progress toward the goals originally undertaken a year and a half ago. I feel confident in stating that we clearly understand the Image Understanding problems in considerably greater depth. I also feel confident that we have made progress in the specific areas of quantitative scene segmentation by clustering, quantitative edge detection and evaluation, and (naturally with the arrival of Dr. Keith Price) have gained a good step toward general symbolic manipulation for the higher levels of many Image Understanding tasks.

Naturally we have also progressed on the traditional front of our expertise, that of Image Processing. The past six months have seen breakthroughs in the areas of variable sampling procedures for image approximations, advances in the a posteriori restoration problem as well as object detection in noisy images. Optical filters for image reconstruction have been designed and the foundations for research in the psychophysical characteristics of the human visual system have been laid.

On the "smart sensor" front considerable effort has been expended in two areas by USC personnel, that of 3 x 3 kernel definition for future sensor implementation, and the study of a real time CCD implementation of an on-board image segmentor. Both these projects represent study efforts for future designs. Naturally Hughes Research Laboratory personnel have also been progressing in the development of test circuitry for the CCD chips under fabrication, and it appears that as of this printing, the Sobel chip is in production and is currently available for testing.

This semi-annual report also includes an overview of the current USCPI laboratory configuration, numerous modifications having been implemented over the past two years. Finally a report of recent Institute Ph.D. dissertations are included as well as the listing of recent Institute personnel publications in the open literature.

\*This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Wright Patterson Air Force Base under Contract F-33615-76-C-1203.

## 2. Image Understanding Projects

This section presents recent results in the research area of Image Understanding. Progress has been achieved in the area of quantifying edge detector parameters by pattern recognition techniques as well as in edge elongation both in monochrome and color scenes. In addition to the above, higher level processes both in symbolic change detection and synthesis of adjacent regions are described. Finally considerable progress has been experienced in the area of automatic scene segmentation from signal processing (bottom up) procedures. The preliminary success of this algorithm is quite encouraging as it utilizes completely unsupervised pattern recognition clustering, feature selection, and cluster optimization techniques without the need for top-down or external guidance. The algorithm is based upon the inherent homogeneity concept of image segments but measured in N-dimensional vector space.

### 2.1 Scene Segmentation by Clustering

Guy Coleman

This project is rapidly advancing toward fruition and represents a bottom up unaided scene segmentation procedure which is based upon homogeneity concepts in N-dimensional vector space. Mathematical pattern recognition, feature selection and clustering techniques are utilized and quantitative evaluations (comparisons) are performed. Because of the success of this project, it is reported on in greater detail in this Image Understanding workshop.

### 2.2 Symbolic Change Analysis

Keith Price

Recent work in image understanding has shown that symbolic techniques can be applied to a large class of images with a variety of change analysis tasks. The system to perform this analysis is now operational at USC (having been developed at CMU). Work is continuing in the areas of: additional use of knowledge in matching, additional task domains, the actual matching function, the use of the change results, change analysis in sequences of images, and the use of these techniques in more general image understanding systems.



### 2.3 Synthesis of Adjacent Regions Erica Rounds

This work describes an algorithm for reconstructing a digital image given the boundary vector lists of relations contained in the image. Permissible topological relations between regions are adjacency and containment. Interior points are assigned to regions on the basis of a small set of "boundary types." These encode the shape of a contour segment connecting three adjacent vertices. The algorithm processes all regions together so that space and time requirements are minimized.

### 2.4 Extension of Boundary Segments in a Multi-Level System Ramakant Nevatia and Kenneth Laws

This section describes continuing efforts in our approach to scene segmentation by edge detection based methods. Obtaining boundaries of objects of interest is of central importance in analysis of a scene. Previously we have described a technique that links local edges detected in an image into larger segments, providing partial boundaries for objects and removing much of the undesired textured background. Extension of such edge segments to yield more complete (longer segments) boundaries is described here.

### 2.5 Detection of Edges in Elongated Neighborhoods Ramakant Nevatia and Peter Chuan

Here we describe a technique for detecting edges that belong to elongated segments. This restriction is expected to provide sensitivity to desired types of edges and not to fine texture or random noise (to which Sobel, Roberts' and Hueckel operators tend to respond). The technique is simply to convolve an image with elongated neighborhoods in various directions. Each convolution gives a value indicating the magnitude of edge in that direction. The maximum value at each point and associated direction are chosen as indicative of edge magnitude and direction at that point.

### 2.6 Color Edge Detection in Scene Segmentation Ramakant Nevatia

A color edge detector, based on the achromatic Hueckel edge operator has been described previously. This report discusses the usefulness of such color edges in scene segmentation in comparison to the use of achromatic edges, and provides an update of the previous results. The concept of edge linking in color space is developed and it is demonstrated that the use of such edges in color aid in building a more robust and reliable system. Further experimentation is required to determine if the improved performance using color is worth the threefold increase in the requirements of storage and computation, at the current costs for these resources.

### 2.7 Calculation of Edge Detector Parameters by Ho-Kashyap William K. Pratt and Ikram Abdou

In previous reports we have formulated edge detection as the classical communication problem of signal detection in the presence of noise. In this work edge detection is discussed as a problem of classifying patterns into two classes (edge and no edge). Many techniques have been developed in pattern recognition to solve this problem. One of them, the Ho-Kashyap algorithm, will be analyzed. The Ho-Kashyap algorithm is briefly reviewed, and the algorithm is then used to find parameters of the Roberts' Operator. Results obtained by these parameters are compared with probabilities of detection and false alarm derived theoretically. The experimental results show quite good correspondence with the theoretical, and suggest the Ho-Kashyap algorithm can be a useful quantitative method for edge detector design.

### 3. Image Processing Projects

This section surveys the progress made in the past six months on various image processing projects. Three new areas are discussed, those of image filtering based on the human visual system, optical filters from digitally constructed kinoforms (holograms) and spatial warp techniques. On-going projects include the estimation of object boundaries in noise, and a posteriori restoration. This latter project has experienced preliminary success in deriving the phase component of the OTF from spatially invariant distortions. Finally one project has reached fruition and completion, that of variable knot splines for image approximation. This technique has led to self-adaptive two-dimensional approximation methods which automatically sense the local activity of a region and apply enough knots (samples) locally to minimize a regional approximation. The technique has applicability in bandwidth compression, image understanding, and particularly in adaptive smart sensing. In the former case, adaptive compressions are available. In the latter case on-board high resolution sensor reduction is possible, and in the image understanding case, the knot density represents a useful feature for higher level processing.

### 3.1 Variable Knot Splines for Image Approximations Harry C. Andrews

This report presents a degree of freedom or information content analysis of images in the context of digital image processing. As such it represents an attempt to quantify the number of truly independent samples one gathers with imaging devices. Variable knot splines are utilized in a two-dimensional approximation theory framework, and sample (pixel) density is assigned according to energy in the two-dimensional fourth difference operators. Quite good adaptive compression and approximation is obtained from these results, as viewed in the accompanying aerial reconnaissance scene. The adaptive nature of the algorithm is evident in the farm regions as compared to the urban scene. (See page 5)



### 3.2 Image Filtering Based on Psychophysical Characteristics of the Human Visual System Charles Hall

In the past decade many physiological and psychophysical experiments have given rise to a fairly sophisticated mathematical nonlinear model. This model has been extended to color perception and is being exercised to test its usefulness in two areas of application: first, as a tool in image compression and second, as providing a space in which useful image quality measures can be quantitatively developed. It is anticipated that rate distortion and other assumptions will become much more realistic in the nonlinear perceptual space developing in these studies.

### 3.3 Optical Filters for Image Reconstruction Alexander A. Sawchuk and Chung-Kai Hsueh

The report discusses the use of a computer plotted hologram as the spatial filter in an incoherent optical system. In the special situation where the hologram contains phase variations only, it is called a kinoform. One problem with the kinoform is that it may not exist for a given impulse response. Iteration methods on the computer are used to obtain a kinoform which has a response very close to the desired one. In addition if we allow the kinoform to have a slow variation in amplitude as well as in phase, then a perfect desired impulse response can be obtained. One application of this system is to give a continuous desampled output from the discrete pixels on a CRT or other discrete image display device.

### 3.4 A Technique for A Posteriori Restoration John Morton

This project is attempting to restore a blurred image with a minimum of a priori knowledge. The only assumptions are a spatially invariant point spread function (PSF) and the extent of the PSF is small compared to the extent of the image. Progress to date includes excellent recursive estimation of the magnitude of the optical transfer function (OTF) of the blur and good recursive estimation of the phase of the OTF. Because the phase of the distortion is considered extremely critical additional effort is being spent on improved phase recovery.

### 3.5 Spatial Warp Interpretation Technique William K. Pratt

Image interpretation consists of a description of a scene, or parts of a scene, based upon some symbolic scene representation. A new technique is described for image interpretation of a segmented image containing perspective views of three-dimensional objects against a fixed background.

### 3.6 Estimation-Detection of Object Boundaries in Noisy Pictures Nasser E. Nahi and Simon Lopez-Mora

Algorithms for successively estimating boundaries have been developed in past research reports. In this present report the problem is formulated under a joint estimation-detection

context with an associated cost function. This framework permits us to obtain an optimal boundary estimation processor that includes a choice for the detector component as well as a procedure for optimal selection of the detection threshold.

## 4. Smart Sensor Projects

Our smart sensor effort is progressing nicely with a division of labor between USCIP personnel and Hughes Research Laboratory personnel. As can be seen from the following, simulations at USC indicated very small adaptive convolving kernels can be quite useful for preprocessing close to the front end of a sensor. In addition, such processes, when implemented near the focal plane, provide potential for reduced subsequent dynamic range requirements in higher level processes. The test facility at HRL is progressing and the Sobel chip seems to be making the usual progress through the variety of production facilities necessary to configure such devices. Similar comments can be applied to the Circuit II, our first attempt at "adaptive on-chip" processing. Finally preliminary efforts are underway to design a real time CCD focal plane image segmentor. This represents our first entry into designing actual image understanding algorithms for potential on-board smart sensor implementation.

### 4.1 Enhancement with 3 x 3 Kernels Harry C. Andrews

More sophisticated modern-day digital image processing has led to the study of adaptive (space-variant) enhancement techniques. Coupled with the ability of both smart sensor and digital refresh technology to implement 3 x 3 convolutions within 1/30 second for 512 x 512 x 8 imagery, it was decided to undertake a study of the power and limitations that such 3 x 3 convolving kernel operations could be utilized to the task of smart sensor two-dimensional signal processing. The underlying theme for this study is the utilization of 3 x 3 kernels for use as control signals to implement both linear and nonlinear as well as spatially invariant and variant (adaptive) signal processing functions in two dimensions. Coupled with this motivation is the fact that USCIP and Hughes Research Laboratories are jointly embarking upon the construction of circuits which would potentially be able to implement these signal processing functions. A large variety of algorithms have been developed for these tasks, and probably those which are the most successful would be labeled as nonlinear.

### 4.2 Real Time Implementation of Image Segmentation Guy Coleman

The segmentation procedure developed elsewhere in this report is currently being used to segment images on a general purpose computer. It is possible to implement this scheme, with some suitable modifications, to segment images in near real time, that is, at television rates.

The study of such a real time system is the subject of this section, and block diagrams are configured and sized for potential implementation.

#### 4.3 CCD Image Processing Circuitry

Graham Nudd, Hughes Research Laboratories

During the period covered by this report we have concentrated our efforts principally on developing the integrated circuits necessary to demonstrate feasibility and to verify our concepts. Two circuits have been selected for implementation, each operating on a  $3 \times 3$  array of picture elements.

The first circuit (Test Circuit I), an implementation of the Sobel Operator for edge detection, is fabricated as a n-channel surface CCD and is designed to operate at 10 MHz rate with accuracy of six bits or better. The detailed design and layout of this circuit has now been completed, and devices should be processed by April 1977.

Test Circuit II contains five separate algorithms; low pass filtering, edge detection, unsharp masking, binarization, and adaptive contrast enhancement. This circuit will be built on a second n-channel test chip, and we hope to have devices processed by mid-year. We anticipate that this chip will be approximately 190 mil x 190 mil, and if there is sufficient area, we will include other test circuits on the same chip. The exact space available for other circuits will not be known until a detailed layout has been completed in the next month or so.

Both circuits are analog implementations which perform arithmetic functions, such as the addition, intensity weightings, and the absolute value operation required in the Sobel, at rates equivalent to 200 MHz. Further, the relatively small size of these circuits offer the possibility of highly parallel operations.

#### 5. Institute Facilities

Recent interest and external visitor pressure has initiated the following report in this section. Essentially due to academic courses, summer short courses, research efforts and general interest in the USC Image Processing Institute, a brief description of the facilities developed to date are reported herein. A bit of the design philosophy as well as user oriented scenarios are presented for the reader to get a better feel for the capabilities (and limitations) currently available at the USCIP. For additional details on the laboratories, please consult the various operating manuals and/or cognizant personnel respectively responsible for the various aspects of the Institute.

#### 6. Recent Ph.D. Dissertations

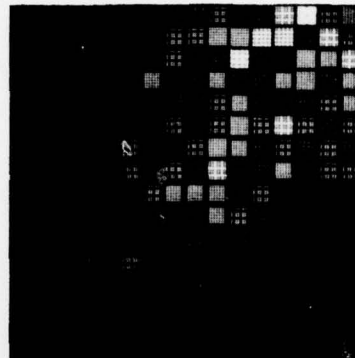
This section includes those dissertations completed since the last reporting period. The one listed here reflects an effort at utilizing two-dimensional approximation theory to much more effectively develop adaptive techniques for efficient image approximations. The results of the research are immediately applicable to high resolution sensors in which channel bandwidth does not permit transmission of the Nyquist resolution everywhere. By on-board variable knot sampling adaptive approximations to the high resolution image are obtained with low dynamic range coefficients. In addition the knot (or sample) density provides a valuable feature for potential on-board segmentation and higher level decision processes.

#### 7. Recent Institute Personnel Publications

This section lists the publications, in the open literature by USCIP personnel. These papers have either appeared, been accepted, or have been submitted for publication in the past six months. 21 such papers are listed.



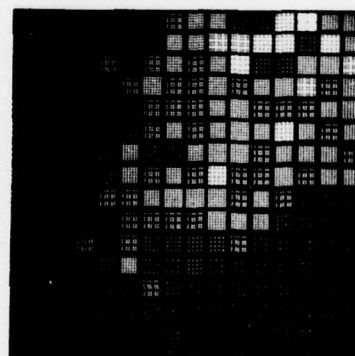
Parameter Reduction = 5.24:1



MSE = 1.2%



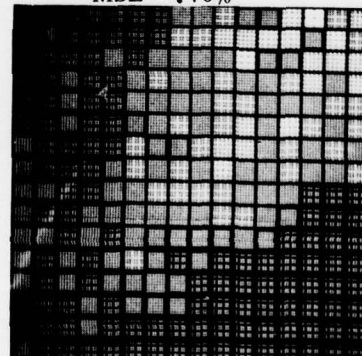
Parameter Reduction = 3.71:1



MSE = .70%



Parameter Reduction = 2.36:1



MSE = .39%

Figure 7. Bicubic Spline Reconstructions and Associated Knot Densities for a Reconnaissance Photograph Using Subregions of Size 16 by 16.